

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Integración de Yunbit CRM con Google Apps

Alberto Manzano Alonso

Tutor: Mario Villar Lozano

Ponente: David Camacho Fernández

MAYO 2016

Integración de Yunbit CRM con Google Apps

AUTOR: Alberto Manzano Alonso
TUTOR: Mario Villar Lozano
PONENTE: David Camacho Fernández

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo de 2016

Resumen

Hoy en día el uso de los servicios de Google está muy extendido entre la sociedad, la disponibilidad de los datos que Google permite junto con la facilidad de uso de sus servicios han hecho que empresas que ofrecen servicios en la nube como Yunbit, inicien proyectos de investigación y desarrollo para dar soporte a tareas de sincronización de datos entre ambas empresas.

El desarrollo de nuevas herramientas que expandan la funcionalidad ofertada desde Yunbit a sus clientes, junto con el propio beneficio de automatizar tareas repetitivas y costosas para los usuarios han sido los motivos que han llevado a la realización de este TFG.

En este documento vamos a ver en detalle el proceso llevado a cabo para la creación de una serie de módulos web junto con su integración en un entorno empresarial actual como posee Yunbit. Se analiza en profundidad el uso de las librerías ofrecidas por Google para el trabajo con sus APIs así como el trabajo a nivel de peticiones HTTP/XML para aquellos servicios que a día de hoy no están presentes en las librerías de Google.

Se realizarán sincronizaciones tanto en ambos sentidos como en un único sentido adecuándose a las necesidades de la empresa y haciendo los módulos tan genéricos como sea posible por si en el futuro se decide personalizar dichas herramientas por necesidades de clientes. Los servicios utilizados en este TFG son Google *Calendar*, Google *Contacts* y Google *Drive* a su vez se ha necesitado hacer uso del servicio Google OAuth para la autenticación de usuarios en plataformas ajenas a Google.

De manera adicional se ha llevado a cabo el desarrollo de una extensión de Google Chrome para la interfaz de Gmail con el fin de mejorar la experiencia de los usuarios con paneles informativos y permitir la vinculación de emails.

Palabras clave

Sincronización, CRM, Google Apps, Google Chrome, Extensión.

Abstract

Nowadays the use of Google services in our society has spread, the availability of the data that Google lets you together with the facility of the use of their services have enable companies, which are currently offering services in the cloud like Yunbit, do research and development projects to synchronize data between both enterprises.

The development of new tools that improve the product offered by Yunbit to their clients together with the advantage to avoid users from doing repetitive and time-consuming tasks have been the main reasons why we have developed the TFG.

In this document we will see in detail the process carried out for the creation of a series of web modules with their integration in a framework like the one Yunbit has. The use of the libraries Google that offers to work with their services, are going to be deeply analysed as well as the use of HTTP/XML requests for those services that are still not present in those Google APIs yet.

Synchronizations will be done in both ways as well as in one way adapting to the company necessities and making those modules as generic as possible in case in the future the clients decide to personalize them. The services used in this TFG are Google Calendar, Google Contacts and Google Drive in turn Google OAuth has been used to authenticate users for those Google outside platforms like Yunbit.

Additionally a new extension of Google Chrome has been developed in order to improve the users experience with informative sidebars and offering a tool to make a connection of emails in Yunbit's platform.

Keywords

Synchronization, CRM, Google Apps, Google Chrome, Extension.

Agradecimientos

Gracias. Gracias en primer lugar a mis padres por apoyarme siempre que lo he necesitado, por motivarme a seguir trabajando y a estudiar lo que realmente me gusta. Gracias a todas aquellas personas que no han parado de darme ánimos, de escucharme cuando me quejaba, de reír conmigo en aquellas tardes en las que no había preocupaciones. Gracias a mis compañeros de Yunbit. Gracias en general a todo el mundo que ha estado presente durante estos últimos 4 años, ya que puedo decir sin miedo a equivocarme que han sido los mejores años de mi vida.

Y gracias a ti por gastar un poquito de tu tiempo para leer este trabajo.

INDICE DE CONTENIDOS

1	Introducción.....	9
1.1	Motivación.....	9
1.2	Objetivos.....	9
1.3	Organización de la memoria.....	10
1.4	Tecnologías empleadas.....	11
2	Autenticación de usuarios y creación de Servicios de Google.	13
2.1	Introducción a Google OAuth 2.0	13
2.2	Análisis de requisitos.....	16
2.2.1	Requisitos funcionales	16
2.2.2	Requisitos no funcionales	18
2.3	Diseño y desarrollo del subsistema	18
2.3.1	Diseño de la base de datos.....	18
2.3.2	Diseño del módulo desarrollado	19
3	Sincronización con Google Calendar.....	23
3.1	Introducción al servicio Google Calendar.....	23
3.2	Análisis de requisitos	25
3.2.1	Requisitos funcionales	25
3.2.2	Requisitos no funcionales	29
3.3	Diseño y desarrollo del subsistema.....	29
3.3.1	Diseño de la base de datos	29
3.3.2	Diseño del módulo desarrollado	32
3.3.2.1	Tarea programada sincronización Google – Yunbit	32
3.3.2.2	Herramienta sincronización Yunbit - Google	33
4	Sincronización con Google Contacts	35

4.1 Introducción al servicio Google Contacts	35
4.2 Análisis de requisitos.....	36
4.2.1 Requisitos funcionales	36
4.2.2 Requisitos no funcionales	37
4.3 Diseño y desarrollo del subsistema	38
4.3.1 Google Contacts API	38
4.3.2 Diseño de la base de datos.....	41
4.3.3 Diseño del módulo desarrollado	44
5 Sincronización con Google Drive	47
5.1 Introducción al servicio Google Drive	47
5.2 Análisis de requisitos.....	47
5.2.1 Requisitos funcionales	47
5.2.2 Requisitos no funcionales	48
5.3 Diseño y desarrollo del subsistema.....	48
5.3.1 Diseño de la base de datos.....	48
5.3.2 Diseño del módulo desarrollado	49
6 Extensión Google Chrome.....	51
6.1 Introducción a las extensiones de Google Chrome	51
6.2 Análisis de requisitos.....	52
6.2.1 Requisitos funcionales	52
6.2.2 Requisitos no funcionales	57
6.3 Diseño y desarrollo de la extensión	57
7 Conclusiones y trabajo futuro	65
Referencias.....	67
Glosario	71
Anexos	i

A	Creación de proyectos en la consola de desarrolladores de Google.....	i
B	Creación y administración de credenciales de proyectos para aplicaciones web.....	i
C	Instalación del proyecto en local.....	i

INDICE DE FIGURAS

FIGURA 1. RELACIÓN ENTRE SUBSISTEMAS	10
FIGURA 2. DIAGRAMA DE SECUENCIA INTERCAMBIO TOKEN.....	14
FIGURA 3. DIAGRAMA ENTIDAD – RELACIÓN USUARIOS	19
FIGURA 4. PORTAL INTRANET PLATAFORMA YUNBIT	20
FIGURA 5. PANEL DE GESTIÓN OAUTH (VACÍO)	21
FIGURA 6. OAUTH GOOGLE	21
FIGURA 7. PANEL DE GESTIÓN OAUTH (LLENO)	22
FIGURA 8. VISTA WEB GOOGLE CALENDAR GENERAL	23
FIGURA 9. VISTA WEB GOOGLE CALENDAR INSERCIÓN	24
FIGURA 10. RELACIÓN SINCRONIZACIÓN YUNBIT – GOOGLE CALENDAR	24
FIGURA 11. DIAGRAMA ENTIDAD – RELACIÓN EVENTOS.....	30
FIGURA 12. PANEL DE GESTIÓN DE CALENDARIOS YUNBIT	33
FIGURA 13. VISTA DE INSERCIÓN DE EVENTO YUNBIT	34
FIGURA 14. VISTA WEB GOOGLE CONTACTS INSERCIÓN.....	35
FIGURA 15. RELACIÓN ENTRE CAPAS GOOGLE CONTACTS 3.0.....	39
FIGURA 16. EJEMPLO DE ENTRADA GDATA	39
FIGURA 17. EJEMPLO OPERACIONES BATCH	41
FIGURA 18. DIAGRAMA ENTIDAD – RELACIÓN USUARIOS (CONTACTOS)	42
FIGURA 19. DIAGRAMA RELACIONAL USUARIOS-PROYECTOS	43
FIGURA 20. DIAGRAMA RELACIONAL USUARIOS-PROPUESTAS	43
FIGURA 21. DIAGRAMA RELACIONAL USUARIOS-EMPRESAS	44
FIGURA 22. HERRAMIENTA DE GESTIÓN DE CONTACTOS	44
FIGURA 23. SINCRONIZACIÓN CONTACTOS (CARGANDO).....	45

FIGURA 24. SINCRONIZACIÓN CONTACTOS (FINALIZADO).....	46
FIGURA 25. CONTACTOS SINCRONIZADOS (GOOGLE CONTACTS).....	46
FIGURA 26. DIAGRAMA ENTIDAD – RELACIÓN DOCUMENTOS.....	49
FIGURA 27. VISTA DE LA HERRAMIENTA PARA LA VINCULACIÓN CON GOOGLE DRIVE	50
FIGURA 28. DISEÑO ICONO SELECCIONABLE EXTENSIÓN	52
FIGURA 29. ACTION PAGE EXTENSIÓN	58
FIGURA 30. ARQUITECTURA DE LA EXTENSIÓN	60
FIGURA 31. DIAGRAMA DE SECUENCIA, INYECCIÓN EXTENSIÓN	60
FIGURA 32. BANDEJA DE CORREO CON PANEL DE USUARIO.....	61
FIGURA 33. BANDEJA DE CORREO CON PANEL DE USUARIO (LOADING).....	62
FIGURA 34. VISUALIZACIÓN DE CORREO CON PANEL	63
FIGURA 35. FORMULARIO VINCULACIÓN EMAIL	64
FIGURA 36. GOOGLE CONSOLE <i>DEVELOPERS</i> (VISTA GENERAL)	I
FIGURA 37. GOOGLE CONSOLE <i>DEVELOPERS</i> (CREDENCIALES).....	I
FIGURA 38. CONFIGURACIÓN DE PANTALLA OAUTH.....	II
FIGURA 39. CONFIGURACIÓN DE CREDENCIALES	III

INDICE DE TABLAS

TABLA 1. ENTIDAD USUARIOS	19
TABLA 2. RELACIÓN CAMPOS GOOGLE EVENT- EVENTOS YUNBIT	31
TABLA 3. CAMPOS CONTACT KIND	41

1 Introducción

1.1 Motivación

Hoy en día, debido a la gran presencia que Google produce en la sociedad y a la cantidad de clientes que actualmente hacen uso de los servicios de Google, se hace vital el poder disponer de la gran cantidad de información que se genera en plataformas externas desde un punto en común.

La posibilidad de disponer de los contactos empresariales en el teléfono móvil o eventos de empresa añade un gran valor a las plataformas de gestión comercial y financiera como Yunbit S.L.[1]. Junto con la posibilidad de expandir la funcionalidad existente a través de los servicios de Google es motivo suficiente para la realización de este TFG.

El hecho de poder trabajar en un entorno de trabajo real junto con la posibilidad de aprender el uso de muchas tecnologías nuevas, han sido motivo suficiente para impulsarme en el desarrollo de este TFG.

1.2 Objetivos

Este proyecto tiene como propósito la unificación de una serie de servicios entre ambas plataformas, Google y Yunbit para facilitar el uso de esta última plataforma a aquellos usuarios que dispongan de cuenta en Google y quieran disponer cierta sincronización entre ambas plataformas extendiendo la funcionalidad existente.

Los objetivos definidos por Yunbit han ido variando frente a la descripción inicial de la propuesta de TFG, a medida que se han ido completando las distintas fases de las cuales se ha compuesto el proyecto.

Los objetivos principales del proyecto son:

- Unificar la información de los usuarios que hacen uso de la plataforma.
- Mantenibilidad de la información sincronizada entre ambas plataformas.
- Disponer de la información relativa a contactos y eventos empresariales en cualquier sitio gracias a una cuenta de Google.
- Facilitar tareas repetitivas y costosas en tiempo a los usuarios.
- Acceso inmediato a elementos almacenados en servidores externos a través de servicios de Google.

A su vez, de manera más específica teniendo en cuenta los servicios con los que se ha trabajado:

- **Eventos:** Sincronización de eventos en ambos sentidos con su correspondiente actualización.
- **Contactos:** Sincronización unidireccional desde Yunbit a Google para disponer de los contactos empresariales en Google Contacts.

- **Documentos:** Sincronización unidireccional desde Google a Yunbit, para disponer de metadatos de los documentos almacenados en Google.
- **Email:** Sincronización unidireccional desde Gmail a Yunbit para disponer de la información de los emails en la plataforma de Yunbit.

1.3 Organización de la memoria

En primer lugar se detallan cada uno de los cuatro módulos realizados para la sincronización con los servicios de Google y por último la extensión de Google Chrome realizada.

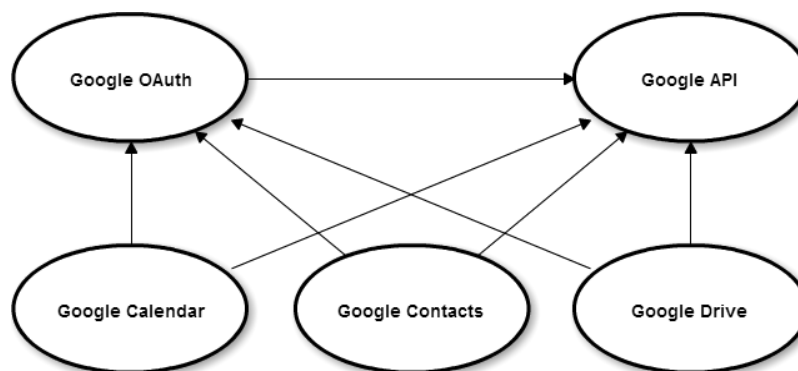


Figura 1 . Relación entre subsistemas

Para simplificar la estructura de la memoria y con el fin de favorecer la lectura continua del documento, se ha optado por fragmentar la estructura de la memoria en cada subsistema implementado, profundizando de manera individual sobre cada subsistema ya que estos no poseen una relación directa entre los mismos a excepción de la dependencia con *Google OAuth*[2] que veremos más adelante.

La estructura que se ha seguido es la siguiente:

1. Explicación del Servicio y su integración en la plataforma de Yunbit.
2. Análisis de requisitos.
 - a. Requisitos funcionales.
 - b. Requisitos no funcionales.
3. Diseño y desarrollo del subsistema.
 - a. Diseño de la base de datos en caso de introducir alguna modificación.
 - b. Diseño del módulo final realizado.

De esta manera cada subsistema es fácilmente localizable junto con las características que lo componen.

En primer lugar se detalla en profundidad el módulo correspondiente al log in de usuarios a nivel de API, Google OAuth, y después en orden secuencial los servicios, calendario, eventos y documentos, para terminar con la extensión de Google Chrome para la interfaz de Gmail.

1.4 Tecnologías empleadas

A continuación se detallan las tecnologías utilizadas durante todo el desarrollo del proyecto junto con las APIs y cualquier otra herramienta utilizada.

Las tecnologías utilizadas en la empresa Yunbit son las siguientes:

- PHP: lenguaje de programación interpretado utilizado para implementar la funcionalidad del lado del servidor.
- Javascript: lenguaje de programación tipado dinámico que soporta la orientación a objetos, utilizado para mejorar la interfaz de las páginas web y hacerlas dinámicas. A su vez ha sido el lenguaje principal utilizado para el desarrollo de la extensión.
 - JQuery[3]: librería utilizada para facilitar la interacción con los elementos DOM de los ficheros HTML desde la parte del cliente.
 - Gmail.js[4]: librería de código abierto utilizada para la interacción con los elementos de la interfaz de Gmail junto con JQuery.
- SQL: Lenguaje de consulta estructurado para bases de datos relacionales, utilizado para la extracción, inserción, modificación y eliminación de datos.
- *Framework* Yunbit: Yunbit ofrece una serie de herramientas para la creación automática de interfaces, facilita la conexión la base de datos y aporta más herramientas para la creación de herramientas en el ecosistema.
- Google Client PHP API[5]: Librería de Google, para el lenguaje de programación PHP que permite trabajar con las peticiones HTTP que se realizan a sus servidores para la administración de los datos de usuario de una manera cómoda. Esta librería se aplica a muchos de los servicios existentes de Google a pesar de ello todavía no han sido portados todos los servicios.

Las herramientas utilizadas para llevar un seguimiento del proyecto así como un control de versiones han sido:

- *Wunderlist*[6]: Herramienta web utilizada para la gestión de tareas.
- SVN: Sistema de control de versiones utilizado actualmente en Yunbit, junto con *Syncro SVN Client*[7] como programa para la gestión visual del repositorio.

Para la implementación del código se ha utilizado como entorno de desarrollo *PhpStorm*[8], es un IDE enfocado al desarrollo web que facilita en gran medida la interacción con distintos servidores.

Otras herramientas utilizadas durante el desarrollo son las siguientes:

- Para el desarrollo local se ha utilizado *WampServer*[9], herramienta que ofrece un servidor Web Apache y un sistema de bases de datos local MySQL, que facilita en gran medida el testeo del producto realizado sin necesidad de utilizar entornos de preproducción o producción.
- Como gestor de bases de datos con interfaz gráfica se ha utilizado *Navicat*[10] que soporta bases de datos de tipo MySQL y conexiones SSH con otros entornos. Para la transferencia masiva de datos desde servidores externos es una herramienta muy potente y fácil de utilizar, de ahí que se haya optado por su uso.

- Para la transferencia de ficheros a otros entornos como el de preproducción se ha utilizado *Filezilla*[11], programa que utiliza el protocolo FTP.
- Para la conexión por consola a otros entornos, se ha utilizado *Putty*[12], como herramienta para la actualización de los ficheros en otros entornos con SVN por consola.
- Google API Console[13]: Herramienta web proporcionada por Google para la gestión de proyectos que hacen uso de servicios de Google a través de APIs. Desde aquí se pueden crear modificar y eliminar proyectos así como consultar estadísticas de uso de los servicios.

2 Autenticación de usuarios y creación de Servicios de Google.

2.1 Introducción a Google OAuth 2.0

Las librerías de Google hacen uso del protocolo OAuth 2.0 para el proceso de autenticación y de autorización. Este protocolo está basado en el uso de un sistema de *tokens* que se intercambian entre las aplicaciones y Google para la autenticación de los usuarios en la plataforma solicitante.

Existen dos tipos de tokens y su obtención depende del escenario en el que nos encontremos así como del tipo de acceso a los datos del usuario que haga la aplicación ya siendo estos *online* u *offline*.

- **Access token:** Código que autoriza la solicitud de un servicio, este código se puede utilizar siempre que no haya caducado, para ello existe una propiedad, *expires_in*, que determina el periodo de tiempo por el que el token será válido para su utilización.
Ejemplo de token de acceso: `{"access_token": "ya29....Ung66lYzg", "expires_in": 3600, "created": 1463049904}`
- **Refresh token:** Código de refresco, que permite solicitar un nuevo token de acceso. Este token no varía con el tiempo y puede renovar tantos tokens de acceso como la aplicación requiera, únicamente se puede obtener si durante el proceso de autorización del usuario se indica que se va a hacer uso de los servicios de manera *offline* es decir, cuando el usuario no esté presente en la aplicación. Véase por ejemplo tareas de mantenimiento de datos o procesos de sincronización cuando el usuario no se encuentra presente en la plataforma como es el caso.

Dependiendo del tipo de aplicación que tengamos la gestión de estos tokens variará así como el proceso de solicitud de permisos a los usuarios, los escenarios existentes son los siguientes:

- **Aplicaciones Web**

Escenario en el que conectamos una aplicación existente que funciona sobre un servidor Web con los servicios de Google, este es el caso de Yunbit.

En este escenario la secuencia de autorización por parte de los usuarios que quieran hacer uso de los servicios, comienza con una redirección a una página de autorización previamente generada a partir de una serie de parámetros, donde el usuario recibe la información relativa a los permisos que la aplicación solicita para posteriormente decidir si se autoriza al uso de dichos servicios en su nombre.

El intercambio de tokens en este escenario funciona de la siguiente manera:

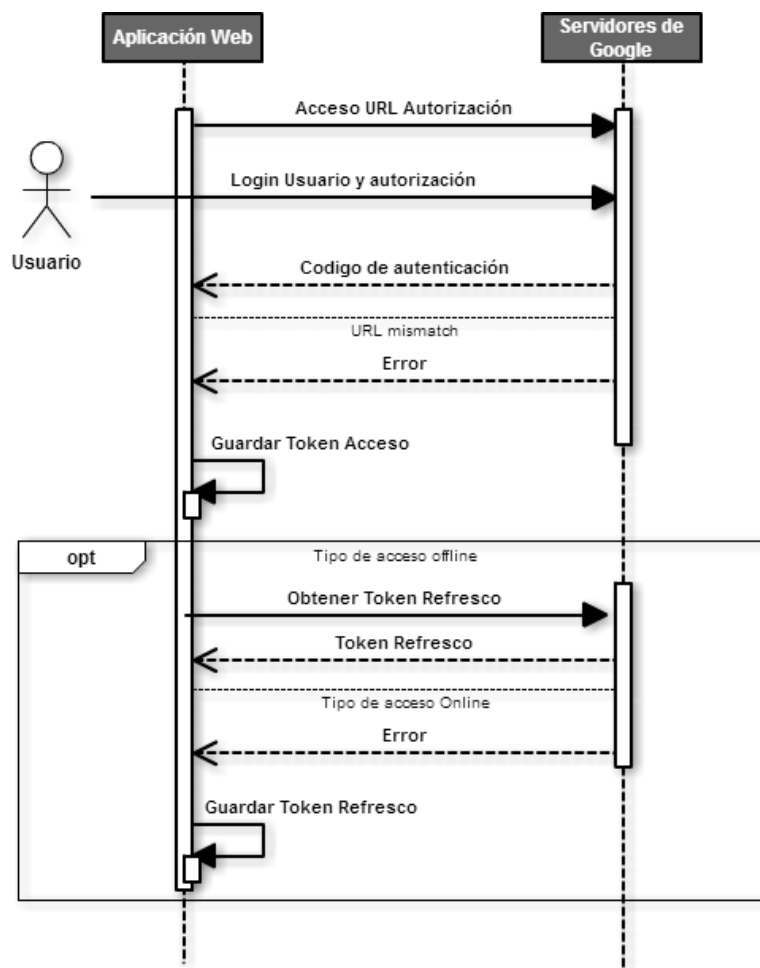


Figura 2. Diagrama de secuencia intercambio Token

Como se puede apreciar en la Figura 2, se pueden producir casos en los que el intercambio de tokens no sea correcto. Estos casos hay que evitarlos para que los usuarios puedan hacer un correcto uso de la herramienta.

A su vez cabe destacar que para el correcto funcionamiento de la aplicación con la librería PHP de Google es necesario cargar previamente un secreto de cliente desde su Google API Console, documento que contiene parámetros de ajuste necesarios para la vinculación entre ambas plataformas.

- **Aplicaciones instaladas**

Similar al caso anterior a excepción de que se utiliza en aplicaciones instaladas en dispositivos, para ello es necesario indicar previamente en qué tipo de dispositivo se va hacer uso de los servicios, ya que Google no tiene manera de identificar la aplicación de manera exacta. De manera adicional informará al usuario de esto durante el proceso de autorización. El sistema de intercambio de tokens es equivalente al de la Figura 2.

- **Aplicaciones cliente Javascript**

Equivalente a las aplicaciones de servidores web, pero con una serie de diferencias, en este caso el secreto de cliente no es secreto por encontrarse embebido en el cliente y ser accesible el código del mismo. Este escenario permite redireccionar a todas las URLs incluyendo así "*http://localhost*", URL que típicamente no se permite en el resto de escenarios.

Por último, no se puede realizar un proceso de autorización incremental, es decir, no se pueden incrementar los permisos autorizados por un cliente sin tener que eliminarlos previamente y crearlos de nuevo con la nueva solicitud.

- **Aplicaciones en dispositivos limitados**

Algunos dispositivos como consolas o cámaras que necesitan hacer uso de APIs de Google, necesitan un intermediario para la fase de autorización introduciendo un código de autenticación proporcionado por Google, por ser un tipo de aplicación que difiere notablemente con el contenido del TFG no se profundiza en sus sistema de intercambio de tokens.

- **Aplicaciones servidor-servidor**

Típicamente aplicaciones web que hagan uso de servicios del tipo Google *Maps*[15], es decir, aplicaciones que no accedan a datos de usuarios si no que hagan uso de una cuenta de Google para obtener acceso a los servicios de Google con sus propios datos, como texto o direcciones.

Doy paso a explicar en mayor profundidad la utilización del protocolo OAuth con aplicaciones Web, por ser la que tiene mayor interés en el proyecto.

El protocolo desde el punto de vista del usuario funciona de la siguiente manera:

1. El usuario que quiere autorizar a la aplicación, accede a una URL proporcionada por la aplicación que dirige a los servidores de Google con una serie de parámetros ya definidos.
2. El usuario da permisos a la aplicación para hacer uso de una serie de servicios en su nombre.
3. El usuario regresa de manera automática a la plataforma inicial trayendo consigo un código de acceso que autoriza a los servicios antes mencionados.

El protocolo en sí, es más complejo y todos los pasos que se realizan sin que el usuario lo vea son los siguientes:

1. Cuando el usuario accede a la aplicación, se establecen los parámetros para la generación de la URL de acceso a los servidores de Google, estableciendo de que servicios va a hacer uso el usuario.
2. Una vez generada la URL y tras el proceso de autorización del usuario, se intercambia el código de autorización por un token de acceso.
 - A El token de acceso se utiliza para realizar las solicitudes de los servicios
 - B Se solicita a los servidores de Google un token de refresco y se almacena en un sistema de almacenaje de datos persistente como una base de datos.

3. El elemento *gclient* ya está completo y se puede empezar a hacer uso de los servicios por parte de la plataforma.

2.2 Análisis de requisitos

En esta sección se presentan los requisitos tanto funcionales como no funcionales del subsistema de autenticación de Google. Para ello se muestran los datos de entrada y de salida correspondientes a cada requisito junto con una breve descripción del funcionamiento del mismo.

2.2.1 Requisitos funcionales

RF 1.1 Selección de servicios a utilizar

Descripción: Se ha de disponer de un sistema de selección de servicios, para determinar cuales se van a utilizar en la plataforma de Yunbit por parte del usuario.

Entrada:

- Seleccionable de tipo *checkbox* que muestra las distintas opciones:
 - Google Calendar
 - Google Contacts
 - Google Drive

Salida: Actualización del campo correspondiente en la base de datos.

RF 1.2 Generación de URL de autenticación

Descripción: Se ha de crear la URL de autenticación con los servidores de Google en base a los servicios que previamente haya seleccionado el usuario (*RF 1.1*).

Entrada:

- Listado de servicios a autorizar
- Nombre de la aplicación
- Tipo de acceso (online u offline)
- URL de redireccionamiento
- Fichero JSON de configuración de la aplicación

Salida: Cadena con la correspondiente URL de autenticación.

RF 1.3 Solicitud de token de refresco

Descripción: Se ha de poder solicitar a Google el token de refresco para actualizar los tokens de acceso en un futuro sin necesidad de solicitar de nuevo permisos al usuario.

Entrada:

- Objeto o instancia ya autenticada con Google OAuth

Salida: Token de refresco de tokens de acceso

RF 1.4 Autenticación de usuario

Descripción: Comprobación de token de autenticación e intercambio por el token de acceso y de refresco (RF 1.3).

Entrada:

- Código de autenticación proporcionado por Google

Salida: Actualización en la base de datos del usuario introduciendo los tokens de acceso y de refresco intercambiado por el de autenticación.

RF 1.5 Obtención de servicios con autorización

Descripción: Una vez que el usuario puede hacer uso de los servicios, necesitamos saber exactamente a qué servicios finalmente ha dado permisos.

Entrada:

- Token de acceso actualizado.

Salida: Listado de servicios autorizados por el usuario.

RF 1.6 Actualización de token de acceso

Descripción: Será necesario comprobar y en caso de ser necesario, actualizar el token de acceso correspondiente al usuario cuando se vaya a hacer uso del mismo.

Entrada:

- Token de refresco
- Objeto o instancia ya autenticada con Google OAuth

Salida: Token de acceso actualizado y actualización del token de acceso en la base de datos.

RF 1.7 Eliminación de permisos

Descripción: Se ha de disponer de una herramienta para eliminar los permisos dados por un usuario a la plataforma Yunbit.

Entrada:

- Objeto o instancia ya autenticada con Google OAuth

Salida: Estado de la operación y eliminación de los tokens almacenados en la base de datos.

2.2.2 Requisitos no funcionales

RNF 1.1

Es necesario que el usuario final no tenga que tener conocimientos acerca del uso de los tokens y el sistema que se produce por debajo, para el usuario el proceso debe de ser tan simple como seleccionar que servicios quiere utilizar y dar permisos a la aplicación.

RNF 1.2

La interfaz que ofrezca estas características al usuario ha de ser fácil de utilizar e intuitiva.

RNF 1.3

El sistema tiene que evitar en la medida de lo posible estar realizando peticiones a los servidores de Google continuamente, por lo que hará uso de un sistema de almacenaje de tokens persistente para los tokens de acceso, una base de datos.

2.3 Diseño y desarrollo del subsistema

2.3.1 Diseño de la base de datos

En primer lugar se muestra un diagrama entidad-relación, Figura 1, con la información necesaria para llevar a cabo el módulo de autenticación con Google.

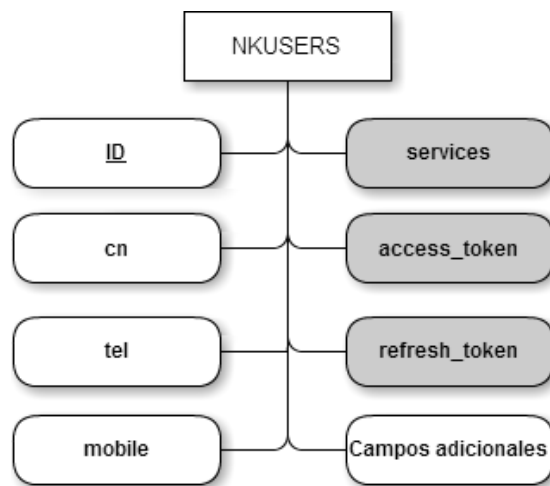


Figura 3. Diagrama Entidad – Relación Usuarios

En esta figura se puede ver que se han introducido tres campos nuevos a la tabla ya existente de la plataforma Yunbit. Estos cambios son necesarios para almacenar la información que veremos a continuación:

NKUSERS

SERVICES	Campo que almacena los servicios a los cuales se quieren dar permisos para la creación de la URL de autenticación.
ACCESS_TOKEN	Token de acceso utilizado en las peticiones a los servicios de la API. Se almacena todo el token obtenido tras la autenticación o refresco.
REFRESH_TOKEN	Token de actualización de tokens de acceso.

Tabla 1. Entidad Usuarios

2.3.2 Diseño del módulo desarrollado

En cuanto al diseño del módulo, se ha optado por aplicar el patrón de diseño *singleton* para la creación del objeto que representa esta clase, ya que no tiene sentido para el sistema que exista más de una instancia de esta clase activa por usuario en un determinado momento.

De esta manera garantizamos que siempre se accede a la misma instancia del objeto en cualquier momento y de manera global.

Este módulo realiza una carga en primer lugar la API de Google para hacer uso del servicio de Google OAuth 2.0 mencionado en el punto 2.1. Tras esto y crear por primera vez una instancia de la clase se procede a la carga del valor SERVICES definido en la base de datos para el usuario activo.

Si dicho campo está vacío no se generará una URL de autorización por cuestiones obvias.

Tras el proceso de autorización del usuario Figura 6, cuando se regresa a la pestaña de "Datos personales" Figura 7, se extrae el código de autenticación de la URL como parámetro GET y se intercambia por los tokens de acceso y refresco almacenando su contenido en la base de datos y asociándolo al usuario que ha realizado la autorización.

Esta clase proporciona una herramienta para que desde cualquier punto de la aplicación que requiera hacer uso de un servicio de Google con un usuario autenticado, pueda utilizarla con un token de acceso activo, ya que de manera automática se comprueba la caducidad de dicho token y en caso de haber expirado se procede a su renovación con el token de refresco que estaba en la base de datos, en caso de que el token de acceso no se encuentre caducado, se continuará trabajando con el sin necesidad de solicitar uno nuevo a Google.

Desde el portal inicial de cualquier plataforma de Yunbit siempre que hayamos iniciado sesión como usuario, dispondremos de la pestaña "Datos personales", pestaña que almacena la información personal relativa al usuario que está activo en la plataforma.

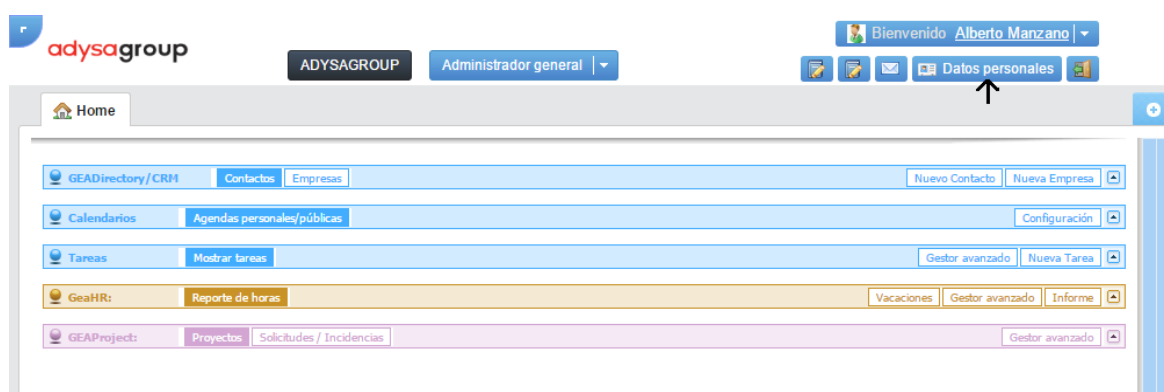


Figura 4. Portal Intranet plataforma Yunbit

La gestión de Google Apps se ha realizado en una pestaña dentro del módulo existente como se puede apreciar en la siguiente Figura 5. Se ha decidido ubicar la gestión de la sincronización con Google Apps en este módulo por unificar todos los servicios que hacen referencia al usuario activo en una misma aplicación. De esta manera los usuarios pueden localizar fácilmente su panel de configuración de Google Apps.


DATOS GENERALES	DATO ADMINISTRATIVO	DATOS IRPF	SOLICITUD DE VACACIONES	HORAS EXTRA	INCAPACIDADES TEMPORALES	DOCUMENTOS ASOCIADOS	HORAS TRABAJADAS	GOOGLE APPS
amanzano@yunbit.es - Alberto Manzano								
Los campos marcados con asterisco * son de obligatoria cumplimentación.								
AJUSTES GOOGLE SYNC								
Token de Refresco								
Sincronización con Servicios de Google		<input type="checkbox"/> Google Calendar <input type="checkbox"/> Google Contacts <input type="checkbox"/> Google drive						
Link de autorización		Autenticación Google						
Eliminar permisos y tokens		<input type="checkbox"/>						
Realizar modificación								




 [TPL - Time: 0.12 s] [Mem: 3.58 Mb] [5]

Figura 5. Panel de gestión OAuth (Vacío)


Estos campos son modificables a excepción del campo “token de refresco” que es puramente informativo, cuando el usuario lo decida realizar una modificación tendrá que confirmar seleccionando en la parte superior “Realizar modificación” de esta manera se efectuarán los cambios de manera definitiva.

A través de esta vista tenemos acceso al hipervínculo en “Link de autorización” que nos lleva directamente a la pantalla de OAuth que se había configurado en el ANEXO B, en esta ocasión vemos la vista como la vería un usuario final, Figura 6.



amanzano@yunbit.es ▼





▼ Yunbit CRM Sync quiere hacer lo siguiente:





Administra tus calendarios.






Ver y administrar los archivos en Google Drive





Administra tus contactos



Al hacer clic en Permitir, autorizas a esta aplicación y a Google a utilizar tu información según lo establecido en sus respectivas condiciones de servicio y políticas de privacidad. Puedes cambiar este y otros [permisos de la cuenta](#) en cualquier momento.

Denegar
Permitir

Figura 6. OAuth Google

Cuando regresamos a la plataforma de Yunbit a través de la URL de redirección indicada previamente, visualizamos la siguiente información ya completa:

DATOS GENERALES	DATO ADMINISTRATIVO	DATOS RPF	SOLICITUD DE VACACIONES	HORAS EXTRA	INCAPACIDADES TEMPORALES	DOCUMENTOS ASOCIADOS	HORAS TRABAJADAS	GOOGLE APPS
amanzano@yunbit.es - Alberto Manzano								
Los campos marcados con asterisco * son de obligatoria cumplimentación.								
AJUSTES GOOGLE SYNC								
Token de Refresco		14pNht5loqVStmEF2yUTxVhFvi-zWSnglpI99ECn_YMEudVK5jSpoR30zcrFq6						
Sincronización con Servicios de Google		[Google Calendar ] [Google Contacts ] [Google drive ]						
Link de autorización		Autenticación Google						
Eliminar permisos y tokens		<input type="checkbox"/>						

Figura 7. Panel de gestión OAuth (Lleno)

A partir de este momento, ya podemos utilizar los servicios de Google que hayamos seleccionado de manera autenticada desde cualquier punto de la aplicación que lo requiera.

3 Sincronización con Google Calendar

3.1 Introducción al servicio Google Calendar

Google Calendar[13], es un servicio gratuito ofrecido por Google, basado en una agenda y calendario electrónico para la gestión de eventos cuya utilización se realiza a través de una interfaz Web. Google Calendar es un servicio incluido de manera automática con la creación de una cuenta de Google.

Desde Google Calendar, los usuarios pueden administrar eventos de manera online, desde su creación hasta su eliminación. Las posibilidades que permite Google Calendar con sus eventos son muy diversas, desde un sistema de notificaciones automático a asistentes, hasta la creación de eventos recurrentes con excepciones de días.

Es una aplicación muy útil por permitir tener sincronizados entre todos nuestros dispositivos que accedan a la misma cuenta de Google, los calendarios personales creados en la aplicación, de manera adicional tiene un sistema de alarmas local para avisar a aquellos dispositivos que tengan un evento próximo de su existencia, dependiendo de si esta opción está activa.

De manera adicional el servicio proporciona herramientas para compartición de eventos notificando a los invitados de la creación, modificación y eliminación del evento junto con un control de asistencia.

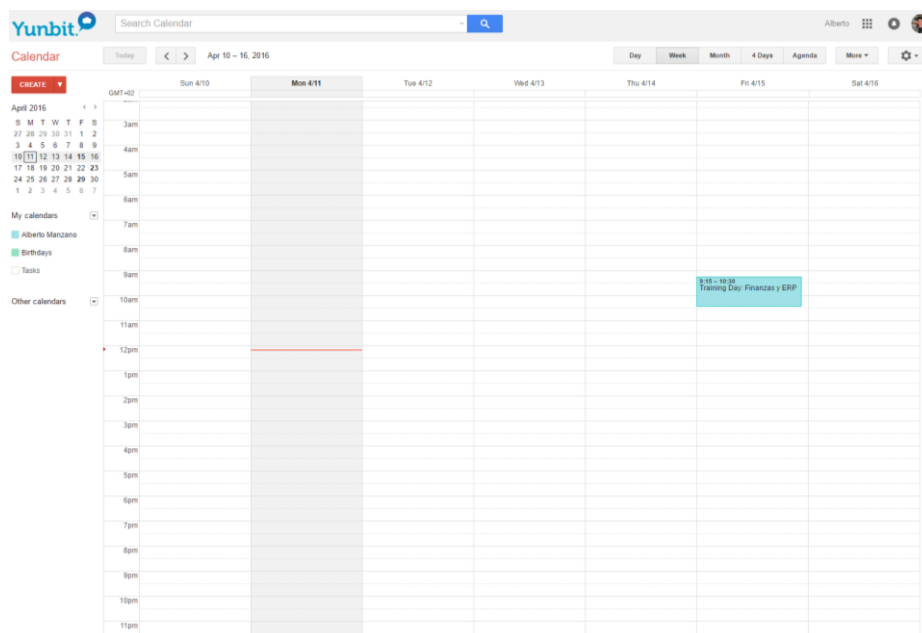


Figura 8. Vista web Google Calendar General

Como vemos a continuación, Figura 9, toda la información relativa al evento se introduce en la creación del mismo a través de distintos formularios.

Por norma general el título junto con las fechas de inicio y de finalización del evento vienen definidas de manera automática, siendo estas modificables de manera manual.

Figura 9. Vista web Google Calendar Inserción

La complejidad de la aplicación varía desde eventos simples con fechas de inicio y de fin, hasta eventos recurrentes con reservas automáticas de espacios. Es una herramienta muy potente que junto con una interfaz atractiva y sencilla de utilizar hace que su uso esté muy extendido entre los usuarios y empresas en la actualidad.

Lo que se pretende con esta herramienta en Yunbit es que los eventos estén sincronizados entre ambas plataformas de manera bidireccional y continua, es decir, si se realizan modificaciones de los eventos por ejemplo, desde un teléfono móvil con la aplicación de Google Calendar, estos cambios deben de producirse de manera similar en la medida que el sistema de gestión de eventos de Yunbit permita y de manera inversa, de Yunbit a Google.

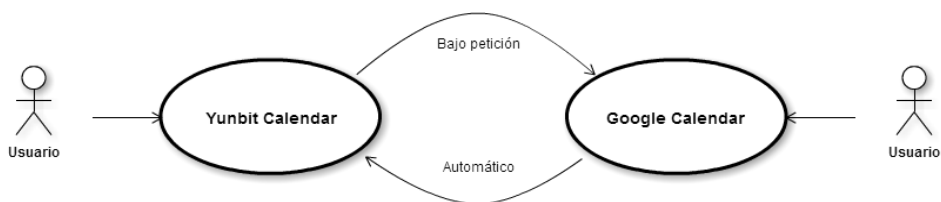


Figura 10. Relación sincronización Yunbit – Google Calendar

Gracias a esta última figura podemos apreciar que el proceso de sincronización desde Yunbit a Google se va a realizar bajo demanda indicando el usuario que esté haciendo uso de la plataforma de Yunbit si desea replicar dichos cambios en Google Calendar.

Por el contrario se requiere que los eventos creados, modificados o eliminados en Google Calendar se sincronicen de manera automática con la plataforma de Yunbit.

Por último indicar que el alcance de este TFG se ha limitado a dar soporte a:

- Eventos individuales entre dos puntos de tiempo.
- Eventos recurrentes básicos.
- Eventos con asistentes.

Ya que para dar soporte a toda la funcionalidad de gestión de eventos que ofrece Google habría que realizar numerosos cambios en el sistema de calendarios actual de Yunbit.

3.2 Análisis de requisitos

En esta sección se presentan los requisitos tanto funcionales como no funcionales del subsistema de sincronización de eventos. Para ello se muestran los datos de entrada y de salida correspondientes a cada requisito junto con una breve descripción del funcionamiento del mismo.

En primer lugar para facilitar la comprensión de los requisitos se especifican aquellos correspondientes a la sincronización de Yunbit a Google y posteriormente los requisitos en el otro sentido.

3.2.1 Requisitos funcionales

RF 2.1 Sincronización de evento

Descripción: Se ha de proporcionar al usuario un sistema para decidir si el evento que está creando, modificando o eliminando de la plataforma de Yunbit va a sincronizarse con Google Calendar.

Entrada:

- *RadioButton* con dos estados dependiendo de si se requiere sincronización o no.

Salida: Replicación de los cambios en Google Calendar o no.

RF 2.2 Crear evento individual

Descripción: Creación de un evento individual entre dos puntos de tiempo con sincronización en Google Calendar desde Yunbit.

Entrada:

- Fecha del evento, fecha de inicio del evento junto con la hora de comienzo.
-

- Fin del evento, fecha de finalización del evento junto con la hora de fin.
- Creador, creador del evento.
- Título del evento.
- Descripción del evento.
- Lugar de evento.
- Privacidad, determina si el evento es público o privado.

Salida: Evento individual creado en la plataforma de Yunbit y en Google Calendar.

RF 2.3 Modificación evento individual

Descripción: Modificación de un evento individual entre dos puntos de tiempo con sincronización en Google Calendar desde Yunbit. Si previamente no existía en la plataforma Google Calendar, se crea por primera vez.

Entrada:

- Fecha del evento, fecha de inicio del evento junto con la hora de comienzo.
- Fin del evento, fecha de finalización del evento junto con la hora de fin.
- Creador, creador del evento.
- Título del evento.
- Descripción del evento.
- Lugar de evento.
- Privacidad, determina si el evento es público o privado.
- Invitados, datos personales de los posibles asistentes al evento

Salida: Evento individual modificado en la plataforma de Yunbit y creado o modificado en Google Calendar en función de su estado previo.

RF 2.4 Creación evento recurrente

Descripción: Creación de un evento recurrente con sincronización en Google Calendar desde Yunbit.

Entrada:

- Fecha del evento, fecha de inicio del evento junto con la hora de comienzo.
- Fin del evento, fecha de finalización del evento junto con la hora de fin.
- Creador, creador del evento.
- Título del evento.
- Descripción del evento.
- Lugar de evento.
- Privacidad, determina si el evento es público o privado.

- Frecuencia de recurrencia, frecuencia diaria, semanal, mensual o anual.
- Fecha de fin de recurrencia.

Salida: Evento recurrente creado en la plataforma de Yunbit y en Google Calendar.

RF 2.5 Modificación evento recurrente

Descripción: Modificación de un evento recurrente con sincronización en Google Calendar desde Yunbit. Si previamente no existía en la plataforma Google Calendar, se crea por primera vez.

Entrada:

- Fecha del evento, fecha de inicio del evento junto con la hora de comienzo.
- Fin del evento, fecha de finalización del evento junto con la hora de fin.
- Creador, creador del evento.
- Título del evento.
- Descripción del evento.
- Lugar de evento.
- Privacidad, determina si el evento es público o privado.
- Invitados, datos personales de los posibles asistentes al evento
- Frecuencia de recurrencia, frecuencia diaria, semanal, mensual o anual.
- Fecha de fin de recurrencia.

Salida: Evento recurrente modificado en la plataforma de Yunbit y creado o modificado en Google Calendar en función de su estado previo.

RF 2.6 Eliminación evento

Descripción: Eliminación del evento que se está mostrando en la plataforma de Yunbit y en Google Calendar en caso de existir.

Entrada:

- Identificador del evento a eliminar

Salida: Desaparición del evento en ambas plataformas.

RF 2.7 Sincronización eventos nuevos de Google Calendar

Descripción: Se requiere la funcionalidad de sincronización de los eventos creados desde Google a Yunbit de manera automática.

Entrada:

- Listado de eventos nuevos a partir de la fecha actual
 - *Id*, identificador único del evento.
 - *Updated*, fecha de actualización.
 - *Created*, fecha de creación.
 - *Start*, fecha de inicio del evento.
 - *End*, fecha de finalización del evento.
 - *Summary*, título del evento.
 - *Description*, descripción del evento.
 - *Place*, lugar del evento.
 - *Attendees*, asistentes al evento.
 - *Status*, estado del evento.
 - *Recurrence*, formato de definición de eventos según el RFC 5545.
- Usuario al que se van a asociar dichos eventos.

Salida: Creación de los eventos en la plataforma de Yunbit.

RF 2.8 Sincronización de eventos modificados de Google Calendar

Descripción: Se requiere poder sincronizar en Yunbit aquellos eventos que han sido modificados desde la plataforma de Google Calendar, con el fin de mantener la consistencia de los datos.

Entrada:

- Listado de eventos modificados a partir de la fecha actual
 - *Id*, identificador único del evento.
 - *Updated*, fecha de actualización.
 - *Created*, fecha de creación.
 - *Start*, fecha de inicio del evento.
 - *End*, fecha de finalización del evento.
 - *Summary*, título del evento.
 - *Description*, descripción del evento.
 - *Place*, lugar del evento.
 - *Attendees*, asistentes al evento.
 - *Status*, estado del evento.
 - *Recurrence*, formato de definición de eventos según el RFC 5545.
- Usuario al que se van a asociar dichos eventos.

Salida: Actualización de los eventos ya existentes en la plataforma de Yunbit.

RF 2.9 Sincronización de eventos eliminados de Google Calendar

Descripción: Se requiere mantener la consistencia de los eventos eliminados en la plataforma de Google Calendar respecto de Yunbit.

Entrada:

- Listado de eventos cancelados a partir de la fecha actual
 - *Id*, identificador único del evento.
- Usuario de la plataforma de Yunbit.

Salida: Eliminación de los datos asociados ha dicho evento.

3.2.2 Requisitos no funcionales

RNF 2.1

Es necesario que la sincronización desde los servidores de Google a la plataforma de Yunbit se realice de manera automática sin interacción del usuario, sincronizando así únicamente los eventos actuales y a futuro.

RNF 2.2

Se ha de dar soporte a la gestión de diferentes zonas horarias con su correspondiente desplazamiento horario.

3.3 Diseño y desarrollo del subsistema

En esta sección vamos a ver en detalle cuales han sido los cambios necesarios en la plataforma existente de Yunbit para llevar a cabo la funcionalidad antes mencionada y su integración.

3.3.1 Diseño de la base de datos

Para adecuarse en esta primera versión a las funcionalidades requeridas no ha sido necesario modificar los campos relativos a eventos ya existentes, si no que únicamente ha sido necesario añadir un nuevo campo de identificación a los eventos con la posibilidad de almacenar el *GID*, identificador único de evento de Google, con aquellos eventos que lo requieran por tener activa la sincronización.

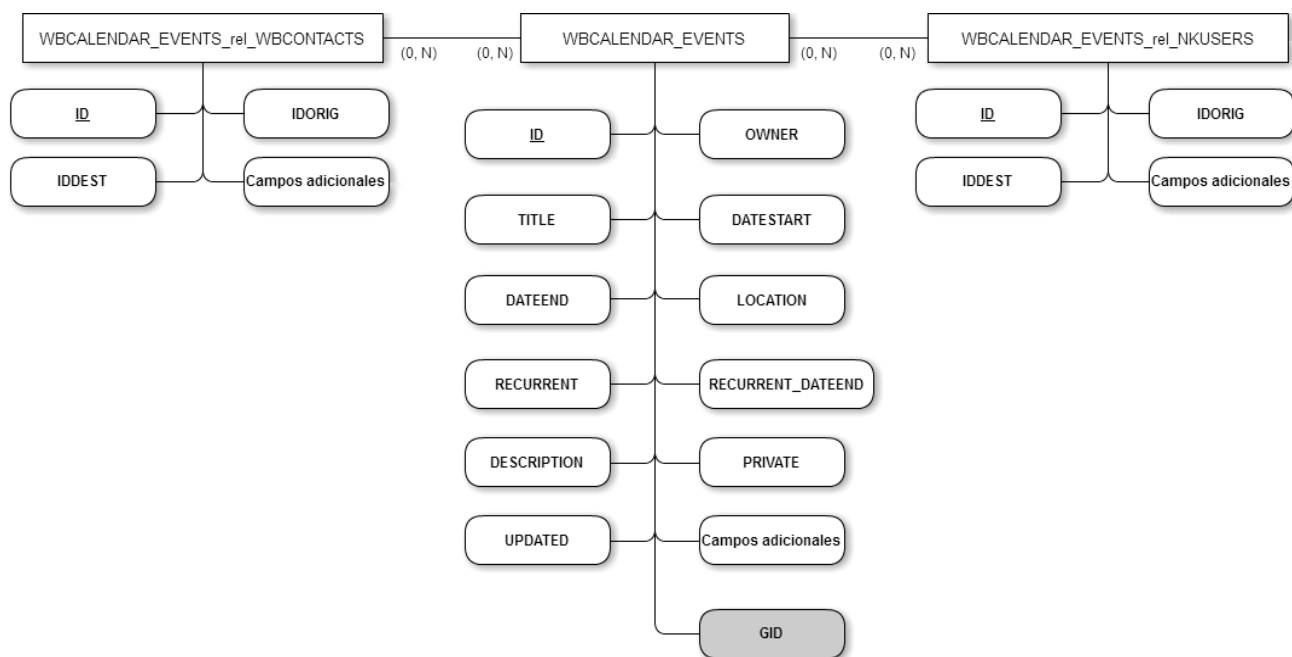


Figura 11. Diagrama Entidad – Relación Eventos

En esta figura podemos ver que el GID es el único campo que necesitamos para extraer información de los eventos de Google o para fácilmente modificar los datos de los mismos en los servidores de Google. A su vez se puede apreciar la relación existente entre contactos y usuarios en los eventos, para las diferencias entre usuarios y contactos se detallan en el punto 4.3.2, en este módulo representan los posibles asistentes que se vinculan con el evento.

A continuación se muestra una tabla de relación entre los campos que Google almacena y los existentes en la plataforma de Yunbit, mostrando únicamente los que tienen relación directa con la funcionalidad implementada en el TFG.

GOOGLE		YUNBIT	
CAMPO	DESCRIPCIÓN	CAMPO	DESCRIPCIÓN
		ID	Identificador único
ID	Identificador único	GID	Identificador de elemento en Google
CREATOR	Creador del evento	OWNER	Usuario propietario del evento

SUMMARY	Título del evento	TITLE	Título del evento
DESCRIPTION	Descripción del evento	DESCRIPTION	Descripción del evento
LOCATION	Lugar del evento	LOCATION	Lugar del evento
UPDATED	Fecha de modificación del evento, formato RFC3339	UPDATED	Fecha de modificación del evento con formato (Y-m-d H-i-s)
START.DATE	Fecha de inicio del evento, formato (yyyy-mm-dd)	DATESTART	Fecha de inicio del evento con formato (Y-m-d H-i-s)
START.TIMEZONE	Zona horaria de la fecha de inicio		
END.DATE	Fecha de finalización del evento, formato (yyyy-mm-dd)	DATEEND	Fecha de finalización del evento con formato (Y-m-d H-i-s)
END.TIMEZONE	Zona horaria de la fecha de finalización		
STATUS	Estado del evento		
VISIBILITY	Visibilidad del evento	PRIVATE	Visibilidad del evento
ATTENDES	Asistentes al evento		
RECURRENCE	Listado de parámetros concatenados que definen el tipo de recurrencia según el RFC 5545	RECURRENT	Tipo de recurrencia según la frecuencia de repetición
		RECURRENT_DATEEND	Fecha de finalización de la recurrencia formato (Y-m-d H-i-s)

Tabla 2. Relación campos Google Event- Eventos Yunbit

Respecto a la relación anterior mencionar que los asistentes a los eventos en Yunbit como se refleja en el diagrama de la Figura 11, son almacenados en dos tablas de la base de datos independientes y no tienen por qué tener relación con los asistentes que se definan en la plataforma de Google Calendar, por lo que se ha decidido que estos sean sincronizados en la única dirección de Yunbit a Google.

Se extrae de la base de datos aquellos o bien contactos o usuarios de las plataformas y se incluyen en el evento de Google con el fin de disponer del sistema de notificaciones de correo de Google.

3.3.2 Diseño del módulo desarrollado

3.3.2.1 Tarea programada sincronización Google – Yunbit

Para llevar a cabo la sincronización de eventos de Yunbit a Google Calendar, se ha optado por crear un proceso recurrente que sincronice aquellas cuentas de los usuarios que tengan activa la sincronización de Google Calendar de manera autónoma.

De tal manera que cada x tiempo, aún por definir por parte de Yunbit, se ejecute un script PHP de manera automática que recorra las bases de datos de los portales de Yunbit en busca de aquellos usuarios que tengan activa la sincronización para actualizar sus calendarios personales.

Este mecanismo de sincronización conlleva a una serie de ventajas y desventajas:

- Los calendarios personales de los usuarios de la plataforma Yunbit tienen sincronizados sus calendarios personales de manera automática.
- Los usuarios no necesitan hacer ningún ajuste o acción adicional para disponer de esta sincronización.
- Si la tarea se demora más de lo normal el usuario no lo percibe.
- Si ocurre un error durante la sincronización de los calendarios, el error es reportado a aquella persona encargada de la tarea programada de manera automática para su posterior solventación.
- De esta manera se pierde la sincronización en tiempo real, de tal manera que no sabremos con seguridad si ambos calendarios están sincronizados en el momento de visualización, ya que el tiempo que pasa entre ejecución y ejecución de la tarea programada, los calendarios pueden verse modificados.

Entrando en detalle con el funcionamiento de la tarea programada. Existe una base de datos que recoge la información necesaria para recorrer cada una de las personalizaciones del portal de Yunbit, de tal manera que la tarea sincroniza los calendarios personales de todos los usuarios de todas las plataformas.

La sincronización se realiza para situaciones diferentes de los eventos:

- **Creación:** Se detecta si en Google Calendar los usuarios disponen de un evento nuevo que no ha sido previamente sincronizado con la plataforma de Yunbit y se procede a su creación con la debida sincronización.

- **Modificación:** Se detecta si un evento ha sido modificado en la plataforma de Google Calendar con una fecha de actualización anterior a la almacenada en la plataforma de Yunbit, en caso de ser así, se extrae la información de dicho evento y se modifican los cambios producidos.
- **Eliminación:** Si se detecta que en la plataforma de Yunbit existen eventos con sincronización con Google Calendar que actualmente ya no existen en dicha plataforma o son considerados como cancelados, estos se actualizan en el sistema de Yunbit con su correspondiente eliminación.

En cada una de estas situaciones se aplica una rutina diferente y cuando finaliza la tarea programada se genera un reporte que es enviado por correo para que los administradores de los sistemas puedan conocer el estado de la tarea y número de eventos sincronizados por usuario.

3.3.2.2 Herramienta sincronización Yunbit - Google

Desde Yunbit, se ha decidido simplificar la integración con la aplicación de calendarios existente, Figura 12, hasta el punto de que en la operación que esté realizando el usuario con marcar un checkbox, Figura 13, dicha operación será adaptada al formato de evento correspondiente para la sincronización con Google Calendar.

De esta manera el usuario tendrá que operar lo menos posible y entenderá que si el checkbox de sincronización con Google Calendar no se encuentra visible es que no ha dado permisos al servicio de Google Calendar en su panel de configuración en datos personales, punto 2.3.2.

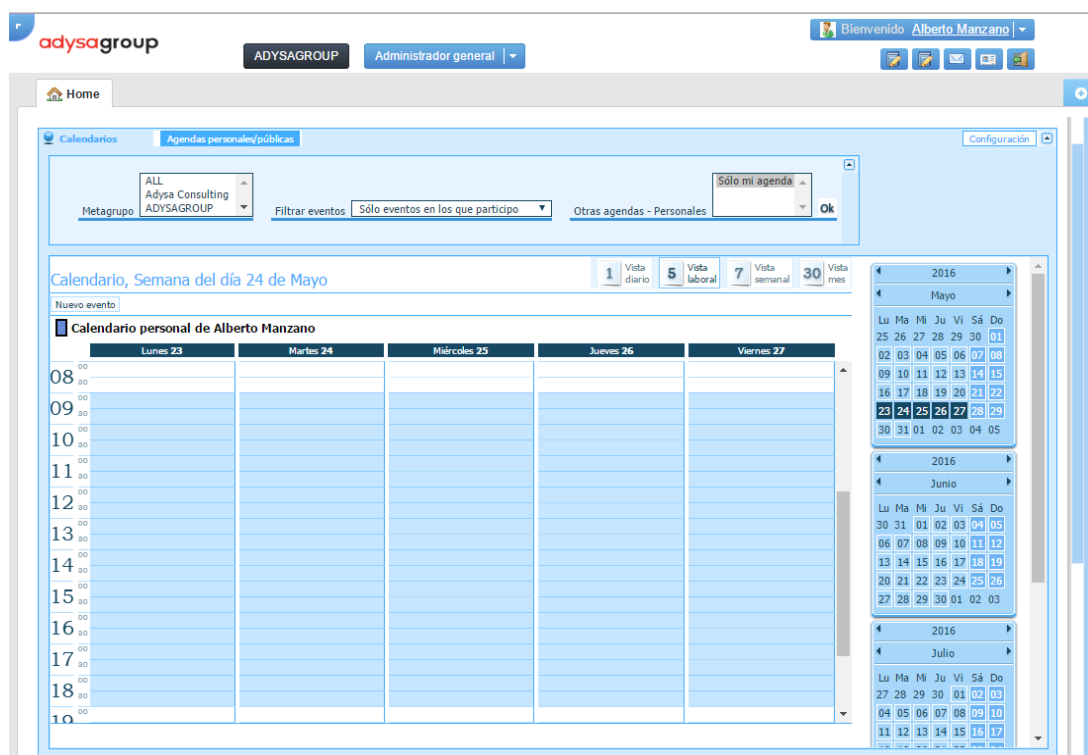


Figura 12. Panel de gestión de calendarios Yunbit

Figura 13. Vista de inserción de evento Yunbit

Durante la inserción, por motivos internos del framework de Yunbit, la relación entre evento y asistentes no se puede realizar por lo que esta opción únicamente está disponible durante la modificación de eventos.

De manera predeterminada se ha decidido que si se opta por la sincronización con Google Calendar, se produzcan notificaciones a los asistentes, tanto de creación, modificación y eliminación en eventos.

El campo de sincronización con Google Calendar únicamente está presente durante la inserción o modificación de eventos y no en listados o visualización de los eventos. Por visualización de eventos se considera a la generación de un listado con los campos del evento en modo de solo lectura, a su vez el modo de listado de eventos es otra funcionalidad ofrecida por Yunbit que permite navegar fácilmente a través de los eventos mostrando aquellos campos que son considerados de mayor importancia.

Como se puede apreciar en la Figura 13 los campos mostrados son los mencionados en el punto 3.3.1 y estos son modificados y ajustados a las necesidades de la API de Google Calendar para que puedan ser gestionados en ambas plataformas. Este proceso de cambio y de detección de la operación que estamos realizando se lleva a cabo en una clase de PHP que requiere que previamente exista un usuario de Google autenticado y que haya dado permisos al servicio de Google Calendar.

4 Sincronización con Google Contacts

4.1 Introducción al servicio Google Contacts

Google Contacts es un servicio proporcionado por Google que consiste en una cartera de contactos integrada en los diversos servicios que Google. Esta cartera de contactos es totalmente gestionable a partir de la interfaz de Google Contacts[16], permite un amplio abanico de posibilidades en cuanto al nivel de personalización que permite de cada contacto así como herramientas para su gestión.

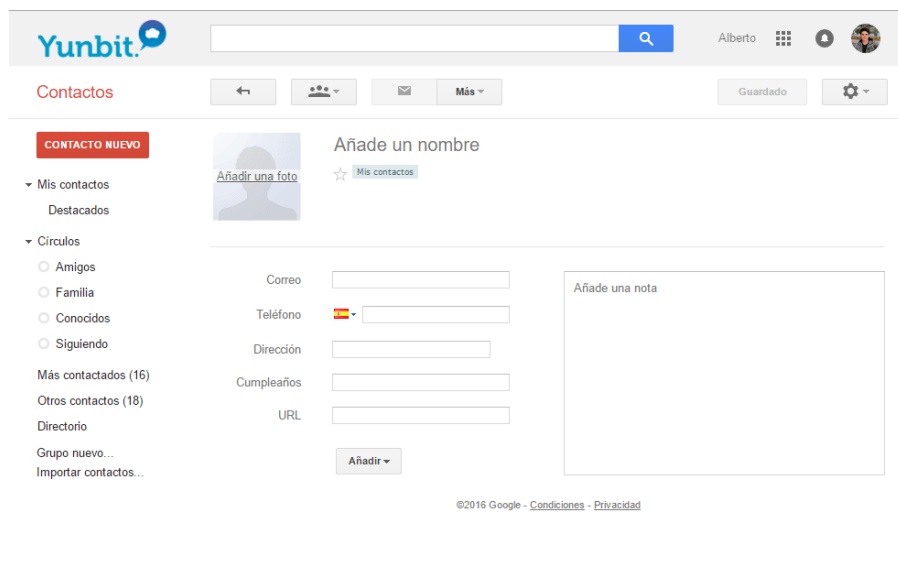
The image shows a screenshot of the Google Contacts web interface. At the top, there's a header with the 'Yunbit' logo, a search bar, and user information 'Alberto'. Below the header, there's a navigation bar with 'Contactos' and several icons. The main content area is titled 'CONTACTO NUEVO' (New Contact). On the left, there's a sidebar with options like 'Mis contactos', 'Destacados', 'Círculos' (with sub-options: Amigos, Familia, Conocidos, Siguiendo), 'Más contactados (16)', 'Otros contactados (18)', 'Directorio', 'Grupo nuevo...', and 'Importar contactos...'. The main form area is titled 'Añade un nombre' (Add a name) and includes a 'Añadir una foto' (Add a photo) button. Below this, there are input fields for 'Correo' (Email), 'Teléfono' (Phone) with a country code dropdown (showing Spain), 'Dirección' (Address), 'Cumpleaños' (Birthday), and 'URL'. There's also a 'Añadir' (Add) button. To the right of these fields is a large text area labeled 'Añade una nota' (Add a note). At the bottom, there's a small copyright notice: '©2016 Google - Condiciones - Privacidad'.

Figura 14. Vista web Google Contacts Inserción

Lo que se quiere obtener a través de la sincronización de los contactos es que cada usuario que haga uso de la plataforma de Yunbit pueda sincronizar de manera unidireccional sus contactos almacenados de la plataforma a Google Contacts.

Antes de continuar cabe aclarar el concepto de contactos existente en el ecosistema de Yunbit. En la plataforma existe una diferenciación entre usuarios y contactos:

- **Usuarios:** Los usuarios son aquellas personas que tienen acceso a las herramientas de la plataforma web.
- **Contactos:** Son entradas de datos personales en las bases de datos de Yunbit que representan personas ajenas a la empresa.

Tanto los usuarios como los contactos tienen datos de carácter personal que se quieren sincronizar y estos no tienen por qué tener una relación entre sí, es decir, existen usuarios que tienen un contacto asociado para extender sus datos personales con los campos reflejados en los contactos al igual que por el contrario existen usuarios que no poseen un contacto asociado.

Se quiere que de manera transparente al usuario, este tenga la opción de sincronizar toda la información de carácter personal sobre las personas que tengan algún tipo de relación con los mismos.

Estos nuevos contactos serán introducidos en el grupo principal que todas las cuentas de Google Contacts disponen, "Mis contactos" y en caso de duplicación por error del usuario, Google pone a disposición de los usuarios una herramienta de eliminación de duplicados muy intuitiva que automáticamente combina los contactos mostrándolos uno a uno con los datos combinados resultantes.

4.2 Análisis de requisitos

En esta sección se presentan los requisitos tanto funcionales como no funcionales del subsistema de sincronización de contactos. Para ello se muestran los datos de entrada y de salida correspondientes a cada requisito junto con una breve descripción del funcionamiento del mismo.

4.2.1 Requisitos funcionales

RF 3.1 Selección de categorías de contactos

Descripción: Se ha de disponer de una herramienta en la interfaz de Yunbit para filtrar que contactos queremos generar en el listado de contactos a sincronizar.

Entrada:

- Checkbox con tres campos a seleccionar:
 - Proyectos relacionados
 - Propuestas comerciales
 - Mis empresas

Salida: Listado de contactos o de usuarios del sistema que cumplen la selección en base al usuario activo en la plataforma.

RF 3.2 Selección de contactos manual

Descripción: Se ha de poder permitir al usuario decidir que usuarios o contactos quiere sincronizar con Google Contacts de manera manual, es decir, uno a uno del listado previamente generado en la interfaz de Yunbit.

Entrada:

- Listado de contactos o usuarios generado en el (RF 3.1).
-

Salida: Listado de contactos o usuarios a sincronizar.

RF 3.3 Selección múltiple de contactos

Descripción: Se ha de ofrecer al usuario una herramienta en la interfaz de Yunbit para seleccionar de un listado todos los contactos disponibles a sincronizar, ya que en algunos casos este listado es demasiado extenso como para realizar una gestión manual.

Entrada:

- Listado de contactos o usuarios generado en el (RF 3.1)

Salida: Listado de contactos o usuarios a sincronizar.

RF 3.4 Vaciar listado múltiple de contactos

Descripción: Se ha de ofrecer al usuario una herramienta en la interfaz de Yunbit para desmarcar todos los contactos de un listado.

Entrada:

- Listado de contactos o usuarios generado en el (RF 3.1)

Salida: Listado vacío de contactos o usuarios a sincronizar.

4.2.2 Requisitos no funcionales

RNF 3.1

Se ha de eliminar la diferencia entre contactos y usuarios de cara al usuario final de la plataforma, únicamente verá un listado con una mezcla entre la información de contactos y usuarios.

RNF 3.2

Es necesario que el tiempo de ejecución no exceda en ningún caso más de 30 segundos de ejecución del servidor para el envío de todas las solicitudes de inserción de los contactos.

La interfaz para la sincronización de datos de contactos ha de ser intuitiva y sencilla de utilizar informando al usuario del estado de la operación de sincronización en todo momento.

En aquellos casos en los que por falta de datos, no se disponga de número de contactos asociados al usuario o contacto, se obtendrá el número correspondiente a su empresa en caso de existir una relación entre ambos, esta relación se detalla en el punto 4.3.2.

4.3 Diseño y desarrollo del subsistema

En esta sección se detallan aquellos aspectos del diseño que se ha llevado a cabo para la implementación de las funcionalidades antes mencionadas.

Comenzando con una explicación de la API para posteriormente detallar como se ha automatizado el proceso de sincronización.

4.3.1 Google Contacts API

Google Contacts es un servicio muy antiguo de Google que todavía no dispone de un módulo gestión de las peticiones HTTP intermediario como otras APIs disponen si no que es necesario el manejo y formación de las entradas XML enviadas a Google de manera manual.

Google poco a poco ha ido actualizando todos sus servicios para que no dependan de la librería Google *Data API*[17], librería que define la estructura de datos de los elementos de Google. Esta librería funciona sobre el protocolo *GData*[18] , actualmente existen muy pocos servicios que sigan utilizándolo.

Google menciona posibilidad de utilizar un nuevo servicio que sí ofrece un módulo dentro de la librería PHP de Google (*People API*), pero este módulo únicamente puede trabajar con la lectura de la cartera de contactos, no permite su por lo que sigue siendo necesario utilizar Google Contacts *API*.

Se da paso a una descripción en profundidad de los distintos niveles que utiliza Google Contacts para el trabajo con la cartera de contactos:

- **Google Data API:** Esta librería sobre la que trabaja Google Contacts es una API o estándar que determina que campos son necesarios para la definición de los objetos a almacenar, Figura 15. La capa de Google Contacts API, determina que datos son necesarios para almacenar objetos de tipo contacto. El protocolo GData así como la estructura y formación de los XML correspondientes a estas operaciones los veremos con detalle a continuación.

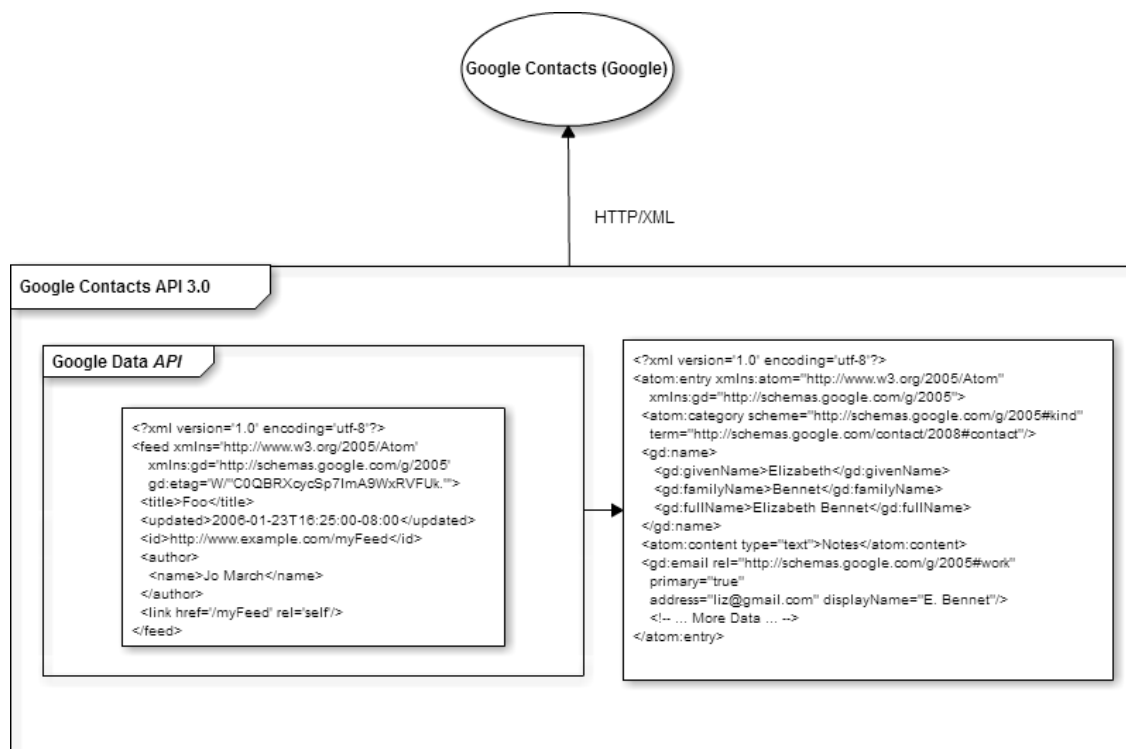


Figura 15. Relación entre capas Google Contacts 3.0

En primer lugar vamos a analizar la formación de un elemento o entrada de Google Data. A continuación en la Figura 16 se muestra un objeto almacenado en los servidores de Google que cumple con el estándar XML de GData.

```

1  <?xml version='1.0' encoding='utf-8'?>
2  <feed xmlns='http://www.w3.org/2005/Atom'
3      xmlns:gd='http://schemas.google.com/g/2005'
4      gd:etag='W/"C0QBRXcycSp7ImA9WxRVFuk."'>
5      <title>Foo</title>
6      <updated>2006-01-23T16:25:00-08:00</updated>
7      <id>http://www.example.com/myFeed</id>
8      <author>
9          <name>Jo March</name>
10     </author>
11     <link href='/myFeed' rel='self'/>
12 </feed>

```

Figura 16. Ejemplo de entrada GData

En este caso vemos la información relativa al usuario que creó dicha entrada, es una entrada vacía que se puede obtener a través de una petición HTTP como:

GET /myFeed

Sin entrar en más detalle sobre los distintos campos que Google Data puede almacenar sobre los objetos, vamos a analizar cuáles son los que definen los datos personales de los contactos en Google Contacts API.

El conjunto de campos que definen los datos personales que se pueden almacenar de los contactos vienen indicados en lo que se denomina *Contact Kind*.

Estos kinds permiten omitir la inserción de información que carece de sentido en determinados contextos, como podría ser en este caso una entrada de un contacto. En estos casos elementos considerados de obligatoria cumplimentación en el estándar *Atom* son omitidos, como:

- **Id**
- **Published:** Fecha de publicación
- **Updated:** Fecha de actualización
- **Author:** Autor de la entrada

Para indicar en la operación que se va a utilizar el esquema o estructura de un objeto de contacto es necesario formar la entrada XML con:

```
< atom: category scheme = "http://schemas.google.com/g/2005#kind"
      term = "http://schemas.google.com/g/2005#contact"/>
```

La estructura del Contact Kind define una serie de parámetros tanto obligatorios como opcionales, a continuación se muestra la composición de estos campos así como todos los que han sido abarcados en el TFG.

CAMPOS

ATOM: CATEGORY	Campo que establece el tipo Contact Kind como estructura a utilizar.
ATOM: TITLE	Nombre del objeto almacenado, típicamente el nombre del contacto.
GD: EMAIL*	Campo opcional múltiple, permite definir múltiples direcciones de correo.
GD: NAME*	Campo opcional, nombre del contacto.

GD: PHONENUMBER*

Campo opcional múltiple, que define números de teléfono.

Tabla 3. Campos Contact Kind

Una vez que disponemos de toda la información necesaria para la creación de los contactos, tan solo hay que escoger la URL adecuada del servicio de Google Contacts API para realizar la operación deseada.

Estas peticiones HTTP van autenticadas gracias al cliente autenticado explicado en la punto 2.1, se realizan a través de la librería de PHP de Google y hasta que no se obtiene la respuesta de cómo ha resultado la operación no continúa la ejecución, lo que es un problema cuando hay subidas iniciales de varios cientos de contactos.

Este problema se ha resuelto haciendo uso de las denominadas operaciones *Batch*. Google ofrece una herramienta para aquellas aplicaciones que necesitan hacer muchas peticiones en poco tiempo, agrupando todas estas peticiones en una sola petición.

Esta petición que engloba a todas las demás tiene un límite de 100 operaciones y es necesario indicar cuál es la operación que se va a realizar en cada entrada o *entry*.

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <feed xmlns='http://www.w3.org/2005/Atom'
3      xmlns:gContact='http://schemas.google.com/contact/2008'
4      xmlns:gd='http://schemas.google.com/g/2005'
5      xmlns:batch='http://schemas.google.com/gdata/batch'>
6      <entry>
7          <batch:id>retrieve</batch:id>
8          <batch:operation type='query' />
9          <id>https://www.google.com/m8/feeds/contacts/default/full/retrieveContactId</id>
10     </entry>
11     <entry>
12         <batch:id>create</batch:id>
13         <batch:operation type='insert' />
14         <category scheme='http://schemas.google.com/g/2005#kind' term='http://schemas.google.com/g/2008#contact' />
15         <!-- Contact Data -->
16     </entry>
17 </feed>
```

Figura 17. Ejemplo Operaciones Batch

Con este sistema se da solución a la limitación del tiempo de ejecución establecida en el (RNF 3.2).

4.3.2 Diseño de la base de datos

A nivel de base de datos, como únicamente se realiza la sincronización unidireccional desde la plataforma Yunbit a Google, tan solo necesitamos almacenar un dato relativo al campo que define de que partes de la aplicación se obtiene el listado de contactos y usuarios a sincronizar.

Siendo este campo similar al campo *SERVICES* definido en el punto 2.3.1 para la definición de los servicios a autorizar.

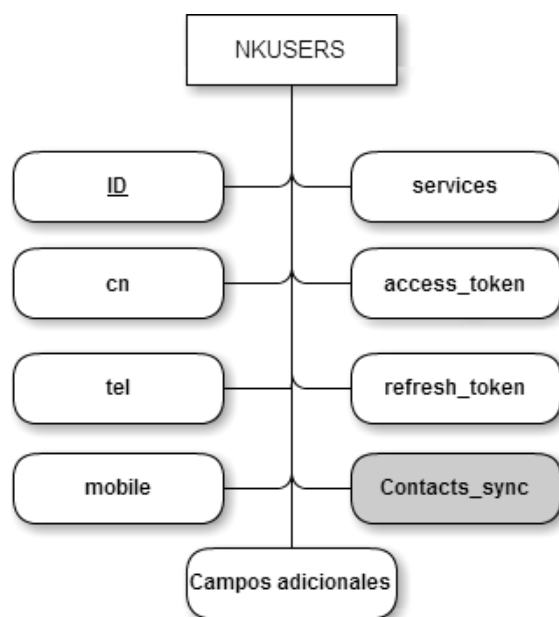


Figura 18. Diagrama Entidad – Relación Usuarios (Contactos)

Para la obtención de los datos de los contactos a sincronizar, en función del campo seleccionado en el checkbox que determina de donde se obtienen los contactos, se siguen los siguientes pasos:

- **Proyectos relacionados:** Existen diversos tipos de usuarios que hacen uso de la plataforma, usuarios de marketing, usuarios desarrolladores, etc. En algunos casos estos son asignados a proyectos dentro de la plataforma. Los datos que se obtienen de esta manera son los datos personales de contactos asociados a los proyectos en los que nos encontramos.
 - **Contactos locales y externos:** Usuarios de la plataforma asignados al proyecto.
 - **Contactos de interés:** Existen contactos relacionados a proyectos que son considerados de interés.

Para entender mejor la relación existente se muestra una porción del diseño de la base de datos existente Figura 19.

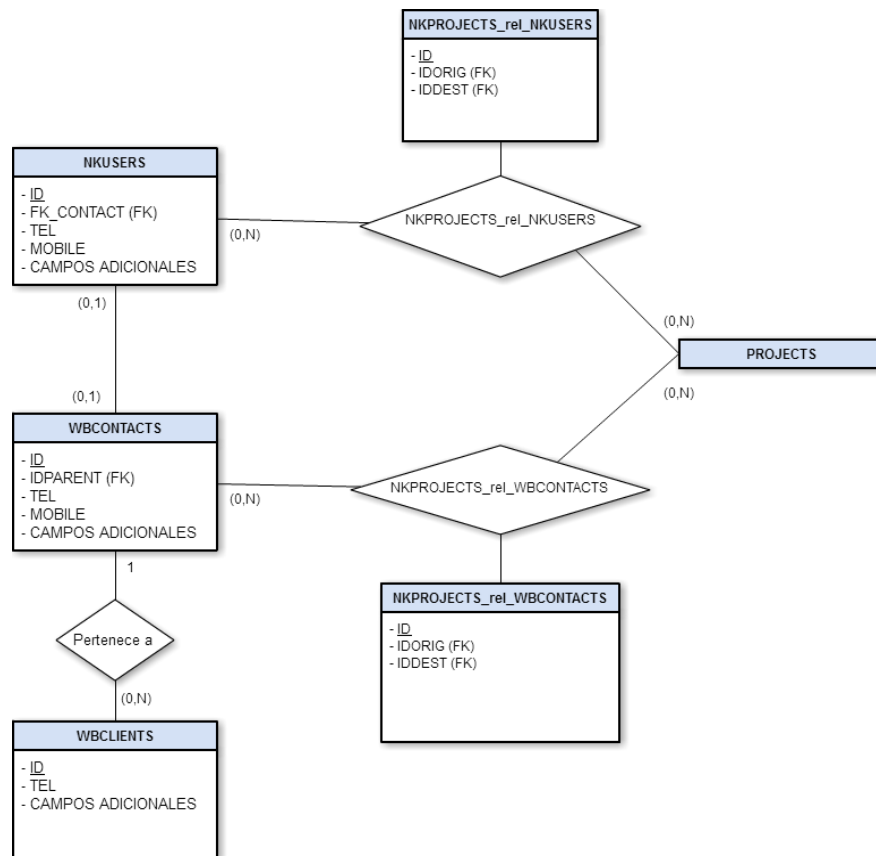


Figura 19. Diagrama relacional Usuarios-Proyectos

- **Propuestas comerciales:** Existe el rol del comercial dentro de la plataforma, en estos casos es interesante obtener los contactos asociados a las propuestas comerciales que realizan.

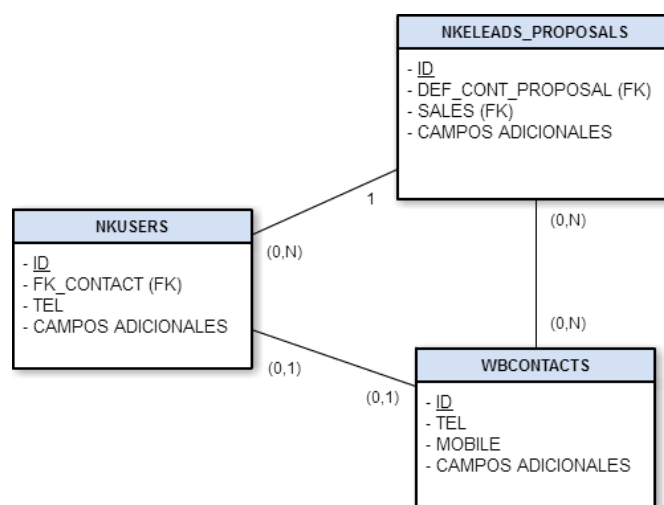


Figura 20. Diagrama relacional Usuarios-Propuestas

- **Mis empresas:** En algunos casos los administradores de los portales pueden representar el rol de propietario de una empresa, en estos casos se obtendrían los contactos de todos aquellos contactos de la empresa propietaria.

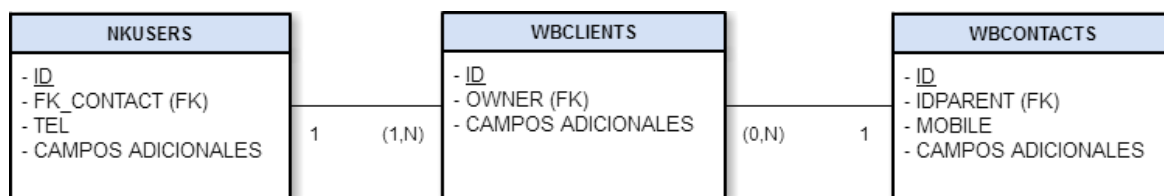


Figura 21. Diagrama relacional Usuarios-Empresas

En todos estos casos partimos siempre desde el mismo campo en común, el identificador del usuario activo en la plataforma y finalmente se juntan los resultados obtenidos de las distintas opciones seleccionadas obteniendo un listado con todos los datos personales ya combinados.

4.3.3 Diseño del módulo desarrollado

Sección que explica el funcionamiento del subsistema así como su integración con las vistas existentes en la plataforma.

Por cuestiones de diseño, se ha decidido que la vista correspondiente a la gestión de los contactos a sincronizar sea la misma que la del punto 2.3.2, para que los usuarios localicen fácilmente esta herramienta.

SINCRONIZACIÓN CON GOOGLE CONTACTS

Sincronizar contactos

[Proyectos relacionados ☒]

[Propuestas comerciales ☐]

[Mis empresas ☒]

Listado de contactos a sincronizar

Sincronizar contactos

Todos

Ninguno

Sync	Nombre completo	Correo electrónico	Teléfono	Móvil
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		648738415
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		648738415
<input type="checkbox"/>	Alberto Manzano	amanzano@adysagroup.com		
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		
<input type="checkbox"/>	Alberto Manzano	amanzano@geanet.es		

Figura 22. Herramienta de gestión de contactos

Desde esta única vista se hace uso de toda la funcionalidad de la herramienta desarrollada.

La primera vez que se entra a la aplicación se ha de definir el estado del seleccionable “Sincronizar Contactos”, con el fin de obtener un listado de contactos como se ha visto en el punto anterior.

Al ser el campo “Listado de contactos a sincronizar” una vista inyectada en un campo, para que su contenido varíe es necesario realizar la modificación del campo “Sincronizar Contactos” previamente para que se genere el listado al recargar la página.

Una vez generado el listado, podemos hacer *scroll* por la vista seleccionando uno a uno los contactos a sincronizar o con los botones de “Todos” o “Ninguno” para interactuar de manera más cómoda por el listado.

Cuando se obtiene el listado final de contactos a sincronizar, se presiona el botón de “Sincronizar contactos” y comienza la transferencia, Figura 23.

SINCRONIZACIÓN CON GOOGLE CONTACTS

Sincronizar contactos de de mis proyectos relacionados

[Proyectos relacionados ☒]

[Propuestas comerciales ☐]

[Mis empresas ☒]

Listado de contactos a sincronizar

Sincronizar contactos

TodosNinguno

Inicio del Proceso de Carga ...

Sync	Nombre completo	Correo electrónico	Teléfono	Móvil
<input checked="" type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		
<input checked="" type="checkbox"/>	Alberto Manzano	amanzano@yunbit.es		648738415
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		
<input checked="" type="checkbox"/>	Alberto Manzano	amanzano@adysagroup.com		
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		
<input type="checkbox"/>	Alberto Manzano	amanzano@geanetondemand.com		

Figura 23. Sincronización Contactos (Cargando)

Cuando la operación es completada como se puede observar en la Figura 24, si se accede a la vista de Google Contacts del usuario que está realizando la sincronización Figura 25, podremos ver que los nuevos contactos se encuentran sincronizados con los datos de la plataforma de Yunbit.

SINCRONIZACIÓN CON GOOGLE CONTACTS

Sincronizar contactos de de mis proyectos relacionados

[Proyectos relacionados ☒]

[Propuestas comerciales ☐]

[Mis empresas ☒]

Listado de contactos a sincronizar

Sincronizar contactos

Todos Ninguno

Contactos sincronizados correctamente

Sync	Nombre completo	Correo electrónico	Teléfono	Móvil
<input checked="" type="checkbox"/>	Alberto Gómez	agomez@geanetondemand.com		686206461
<input checked="" type="checkbox"/>	Alberto Manzano	amanzano@yunbit.es		648738415
<input type="checkbox"/>	Alfonso Manzano	afmanzo@geanetondemand.com		686206461
<input checked="" type="checkbox"/>	Alfonso Fernández de Mesa	afernandez@adysagroup.com		
<input type="checkbox"/>	Alfonso Manzano	amanzano@geanetondemand.com		686206461
<input type="checkbox"/>	Alfonso Manzano	amanzano@geanetondemand.com		686206461

Figura 24. Sincronización Contactos (Finalizado)

Yunbit

Alberto

Contacts

+ ▼

More ▼

Alberto - Alfonso

< > ⚙

NEW CONTACT

▼ My Contacts (3)

Starred

▼ Circles

○ Amigos

○ Familia

○ Conocidos

○ Siguiendo

Most Contacted (16)

Other Contacts (18)

Directory

New Group...

Import Contacts...

☐

☆

Alberto Gómez

agomez@geanetond...

686206461

☐

☆

Alberto Manzano

amanzano@yunb...

(+2) 648738415

☐

☆

Alfonso Fernández d...

afernandez@adysagr...

©2016 Google - [Terms](#) - [Privacy](#)

Figura 25. Contactos sincronizados (Google Contacts)

5 Sincronización con Google Drive

5.1 Introducción al servicio Google Drive

Google Drive es un servicio que ofrece un repositorio de ficheros online. Este servicio permite a los usuarios almacenar elementos, compartirlos y editarlos a través de una interfaz web.

Su principal uso en el ámbito empresarial es la compartición de documentos y la posibilidad de no almacenar dichos documentos en los propios servidores de la empresa.

Desde Yunbit se quiere disponer de una herramienta para mantener una vinculación entre las entradas de sus servidores que hacen referencia a documentos y los documentos en sí almacenados en Google Drive.

Google Drive es un servicio que se mantiene actualizado y actualmente se dispone de un módulo en la API de Google para el trabajo con el repositorio documental de manera cómoda desde PHP. Tan solo es necesario tener un cliente de Google que haya autorizado el uso de dicho servicio a través de APIs.

5.2 Análisis de requisitos

En esta sección se presentan los requisitos tanto funcionales como no funcionales del subsistema de sincronización con el repositorio documental Google Drive. Para ello se muestran los datos de entrada y de salida correspondientes a cada requisito junto con una breve descripción del funcionamiento del mismo.

5.2.1 Requisitos funcionales

RF 4.1 Visualización de árbol de ficheros Google Drive

Descripción: Desde la herramienta de administración de documentos de Yunbit, se necesita visualizar el árbol de contenidos de Google Drive para seleccionar con que elemento se va a realizar la vinculación.

Entrada:

- Listado de elementos en Google Drive desde la raíz sin carpetas.

Salida: Representación de los elementos en Google Drive.

Descripción: Una vez que se conoce que fichero de Google Drive se quiere vincular, se ha de traer la información relativa a dicho elemento desde Google Drive a la plataforma Yunbit.

Entrada:

- Identificador de elemento en Google Drive

Salida: Modificación en la base de datos de Yunbit con:

- Link de previsualización
- Link de descarga

5.2.2 Requisitos no funcionales

El módulo ha de tener una interfaz sencilla de comprender y que visualmente no rompa con el diseño de la plataforma actual.

5.3 Diseño y desarrollo del subsistema

En esta sección se detallan aquellos aspectos del diseño que se han llevado a cabo para la implementación de las funcionalidades antes mencionadas.

Comenzando con una explicación de la API para posteriormente detallar como se ha automatizado el proceso de sincronización.

5.3.1 Diseño de la base de datos

En este caso al ser la sincronización unidireccional y una única vez cuando se crea el documento o cuando este es modificado, tan solo es necesario modificar la estructura actual de la plataforma Yunbit añadiendo los nuevos campos que se van a almacenar y a mostrar a los usuarios.

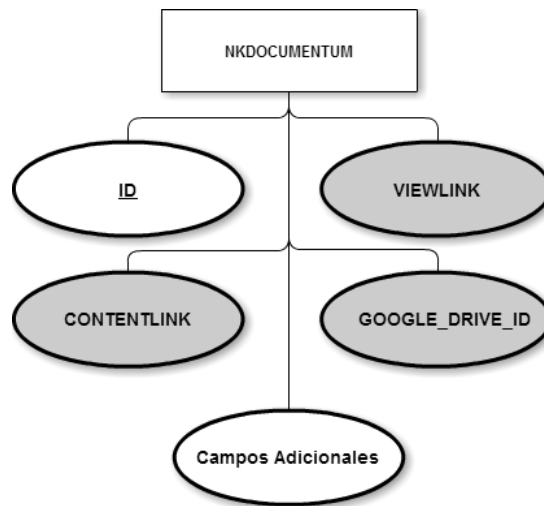


Figura 26. Diagrama Entidad – Relación Documentos

Los campos introducidos son los siguientes:

- **GOOGLE_DRIVE_ID:** Identificador de elemento en Google Drive, necesario para la obtención del resto de metadatos.
- **VIEWLINK:** Hace referencia al link de previsualización que Google proporciona en algunos casos cuando el fichero solicitado dispone de una visualización en red, previa a la descarga.
- **CONTENTLINK:** Link de descarga del fichero existente en todos los elementos de Google Drive que no representen carpetas.

5.3.2 Diseño del módulo desarrollado

Se ha introducido 3 campos nuevos durante la modificación de documentos en la plataforma de Yunbit, por limitaciones del framework estos campos no son incluidos durante la inserción de documentos en la plataforma.

La sección de Google drive consta de los siguientes campos:

- Google Drive: Árbol de documentos
- Google Drive previsualización
- Google Drive link descarga

Campos que únicamente son mostrados al usuario en caso de tener autorizado el uso de Google drive en el panel de información personal visto en el punto 2.3.2.

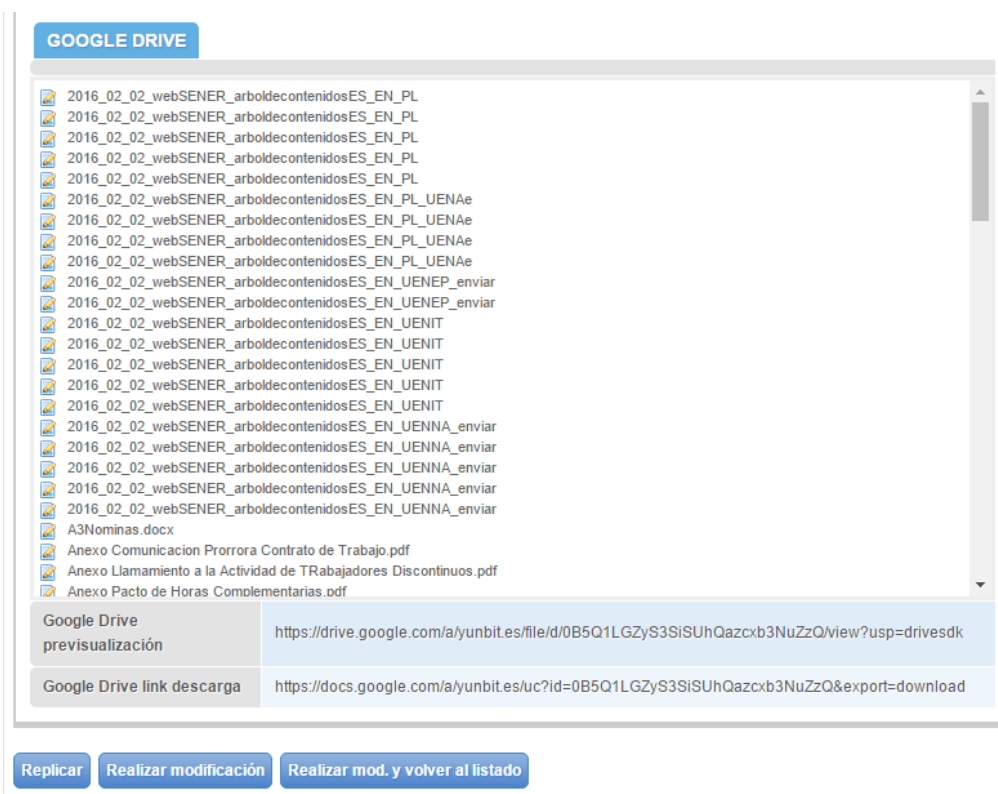


Figura 27. Vista de la herramienta para la vinculación con Google Drive

En el caso de la Figura 27, estamos ante un documento de la plataforma Yunbit, que está siendo modificado, por ello se muestran los campos nuevos, permitiendo seleccionar al usuario un nuevo documento del repositorio de Google Drive con el que vincular el documento.

Cuando se selecciona el documento del árbol de Google Drive, y se realiza la modificación se extraen los datos y se insertan o modifican en la base de datos, de tal manera que durante la visualización del documento o la modificación se muestran los datos previos.

Ambos campos, tanto el link de previsualización como el de descarga únicamente de lectura.

6 Extensión Google Chrome

Con el desarrollo de una extensión de Google Chrome se espera facilitar la sincronización de los correos entre ambas plataformas, de tal manera que la integración de una serie de herramientas nuevas en la interfaz de Gmail sea lo más cómodo e intuitivo para los usuarios.

Para ello se ha creado una extensión de Google Chrome que modifica la vista actual del panel de correo Gmail incluyendo a grandes rasgos dos nuevas funcionalidades.

- Extracción de datos de Yunbit y visualización dentro de la interfaz de Gmail.
- Envío y manejo de datos desde Gmail para la sincronización de emails.

Estas dos funcionalidades se espera que interfieran lo menos posible con la interfaz de Gmail y sean fáciles de utilizar así como de instalar.

Por la falta de tiempo así como de experiencia en el desarrollo de servicios REST en la plataforma Yunbit se ha desarrollado un prototipo no funcional que trabaje con datos propios. En esta sección se tratarán aspectos como cuál ha sido el análisis y diseño de la arquitectura final de la extensión así como el proceso de integración en la interfaz de Gmail a falta de la implementación de dichos servicios en un futuro. Por lo que los datos mostrados en los diseños son ficticios y no tienen relación directa con los datos almacenados en Yunbit.

6.1 Introducción a las extensiones de Google Chrome

Google ofrece a los desarrolladores un amplio abanico de posibilidades a mezclar para la composición de las extensiones en función de cuales sean sus necesidades, disponiendo de aplicaciones instaladas, apps, extensiones y dentro de cada categoría diversas herramientas.

En este proyecto se ha optado por el desarrollo de una extensión de Google Chrome por ser la que mejor se adapta a las necesidades de los usuarios y así requerirlo por parte de Yunbit, para ello vamos a ver que es una extensión de Google Chrome y cuáles son las características que la definen.

Una extensión de Google Chrome es un conjunto de ficheros (HTML, CSS, Javascript, etc.) que añaden funcionalidad al navegador Google Chrome.

En este caso no se quería crear de aplicaciones externas al navegador Google Chrome y ya que el uso de las extensiones de Google Chrome está muy extendido, la integración de una serie de herramientas que modifiquen la interfaz de Gmail ha sido la solución tomada.

Las extensiones de Google Chrome poseen una serie de características comunes entre todas ellas por lo que vamos a analizar una a una cuales son:

- **Interfaz de usuario:** Todas las extensiones de Google Chrome tienen que tener un icono para dar conocimiento a los usuarios de que estas extensiones están habilitadas, estos iconos son mostrados en el lateral de la barra de navegación y en función de su color podemos saber si se están ejecutando en la página web que estamos visualizando, si están realizando tareas en segundo plano o por el contrario no están ejecutándose.

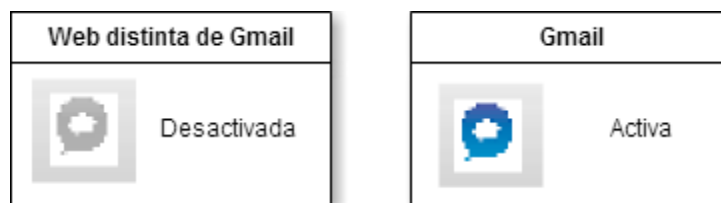


Figura 28. Diseño icono seleccionable extensión

- **Ficheros:** Existe un fichero de configuración que todas las extensiones poseen llamado *manifest* este fichero almacena las características de la extensión definiendo los permisos, recursos y servicios de los que va a hacer uso la extensión así como reglas de seguridad que se detallan en mayor profundidad en el punto 6.3. El resto de ficheros son opcionales y dependen de la funcionalidad de la extensión.

6.2 Análisis de requisitos

En esta sección se presentan los requisitos tanto funcionales como no funcionales de la extensión de Google Chrome. Para ello se muestran los datos de entrada y de salida correspondientes a cada requisito junto con una breve descripción del funcionamiento del mismo.

6.2.1 Requisitos funcionales

RF 5.1 Creación de credenciales de usuario

Descripción: Es necesario que la extensión almacene de manera segura las credenciales de acceso a la plataforma de Yunbit correspondiente para poder hacer uso de los servicios de Yunbit. Estos servicios de Yunbit pueden funcionar o bien a partir de una contraseña o bien mediante el uso de un token previamente generado en la plataforma de Yunbit.

Entrada:

- Email, email de identificación obligatorio de la plataforma de Yunbit.
- Password, contraseña del email.
- Token, token previamente generado en la plataforma Yunbit.

Salida: Actualización del sistema de almacenaje de la extensión con los datos. Inyección de la extensión en Gmail.

RF 5.2 Modificación de credenciales de usuario

Descripción: Es necesario disponer de un sistema de modificación de credenciales por su posible modificación en el futuro.

Entrada:

- Email, email de identificación obligatorio de la plataforma de Yunbit.
- Password, contraseña del email.
- Token, token previamente generado en la plataforma Yunbit.

Salida: Modificación de las credenciales guardadas en la extensión con los datos nuevos.

RF 5.3 Obtención de las tareas de usuario

Descripción: Como aspecto informativo se ha de mostrar al usuario la lista de tareas existentes de la plataforma Yunbit en la interfaz de Gmail.

Entrada:

- Email, email del usuario que realiza la solicitud de obtención de sus tareas.
- Password o Token, necesario para autorizar la llamada a los servicios REST.

Salida: Listado de tareas del usuario solicitante y su visualización por pantalla.

RF 5.4 Obtención de los eventos de usuario

Descripción: Como aspecto informativo se ha de mostrar al usuario el listado de eventos futuros existentes de la plataforma Yunbit en la interfaz de Gmail.

Entrada:

- Email, email del usuario que realiza la solicitud de obtención de los eventos.
- Password o Token, necesario para autorizar la llamada a los servicios REST.

Salida: Listado de eventos del usuario solicitante y su visualización por pantalla.

RF 5.5 Obtención de las últimas actividades realizadas de usuario

Descripción: Como aspecto informativo se ha de mostrar al usuario un listado con las últimas actividades realizadas de la plataforma Yunbit en la interfaz de Gmail.

Entrada:

- Email, email del usuario que realiza la solicitud de obtención de sus últimas actividades realizadas.
- Password o Token, necesario para autorizar la llamada a los servicios REST.

Salida: Listado de actividades recientes por el usuario solicitante y su visualización por pantalla.

RF 5.6 Obtención de los datos de un contacto

Descripción: Cuando nos encontramos dentro de un email es necesario obtener la información relativa a dicha dirección de correo de la plataforma Yunbit, más concretamente los datos personales de dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de datos personales relativos al email mostrado.

- Nombre
- Teléfono
- Email

RF 5.7 Obtención de las tareas de un contacto

Descripción: Cuando nos encontramos dentro de un email es necesario obtener la información relativa a dicha dirección de correo de la plataforma Yunbit, más concretamente las tareas pendientes a realizar con dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de tareas pendientes relativas al email mostrado.

RF 5.8 Obtención de los eventos de un contacto

Descripción: Cuando nos encontramos dentro de un email es necesario obtener la información relativa a dicha dirección de correo de la plataforma Yunbit, más concretamente los eventos que posee dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de eventos futuros existentes en Yunbit sobre el email mostrado.

RF 5.9 Obtención de las oportunidades de un contacto

Descripción: Cuando nos encontramos dentro de un email es necesario obtener la información relativa a dicha dirección de correo de la plataforma Yunbit, más concretamente las oportunidades de marketing relacionadas con dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de oportunidades relativas al email mostrado.

RF 5.10 Sincronización de un email con Yunbit

Descripción: Cuando nos encontramos dentro de un email se ha de disponer de una herramienta de envío del email mostrado a la plataforma de Yunbit para disponer de una relación o asociación entre contactos y emails.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.
- Cuerpo del email, contenido del email.
- Formulario adicional de vinculación.
 - Tipo de acción
 - Contacto relacionado

- Propuesta relacionada
- Cuenta relacionada

Salida: Solicitud de inserción en la plataforma de Yunbit del email junto con su relación con las correspondientes partes de la plataforma.

RF 5.11 Obtención de los tipos de acción

Descripción: Dentro del formulario de vinculación de un email con la plataforma de Yunbit, es necesario seleccionar el tipo de acción con la que se va a realizar la vinculación, este listado es obtenido de Yunbit.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.

Salida: Listado de tipos de acción de la plataforma Yunbit.

RF 5.12 Obtención de contactos relacionados a contactos

Descripción: Dentro del formulario de vinculación de un email con la plataforma de Yunbit, se ofrece la posibilidad de relacionar el email con otro contacto de la plataforma de Yunbit.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de contactos de la plataforma Yunbit relacionados a su vez con el contacto del email a vincular.

RF 5.13 Obtención de propuestas relacionadas a contactos

Descripción: Dentro del formulario de vinculación de un email con la plataforma de Yunbit, se ofrece la posibilidad de relacionar el email con una propuesta relacionada con dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.

- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de propuestas de la plataforma Yunbit relacionadas a su vez con el contacto del email a vincular.

RF 5.14 Obtención de cuentas relacionadas a contactos

Descripción: Dentro del formulario de vinculación de un email con la plataforma de Yunbit, se ofrece la posibilidad de relacionar el email con una cuenta relacionada con dicho contacto.

Entrada:

- Email, email del usuario que realiza la solicitud.
- Password o Token, necesario para autorizar la llamada a los servicios REST.
- Email, email del sujeto que ha enviado el email en visualización.

Salida: Listado de cuentas de la plataforma Yunbit relacionadas a su vez con el contacto del email a vincular.

6.2.2 Requisitos no funcionales

RNF 5.1

La interfaz de la extensión ha de ser fácil de utilizar por usuarios no experimentados en el uso de extensiones de Google Chrome

RNF 5.2

El usuario de la extensión ha de ser informado continuamente de una manera visual sobre el estado de las operaciones llevadas a cabo. Ya sean operaciones informativas como mostrar información de la plataforma de Yunbit o el envío de emails.

6.3 Diseño y desarrollo de la extensión

Para el diseño de la arquitectura de la extensión me he basado en extensiones actuales que tienen usos similares como pueden ser las extensiones de Ebsta para Salesforce[20], Wunderlist, Strike[21], Right in box[22] y El camelizer[23].

Tras el análisis de estas extensiones y a la documentación que Google proporciona para el desarrollo de extensiones se ha definido la siguiente arquitectura:

- **Action Page:** Como hemos visto anteriormente Google muestra un icono de manera predeterminada para cada una de las extensiones habilitadas, en este caso el hecho de poseer un action page permite activar o desactivar la funcionalidad de la extensión a partir del filtro de la URL para que únicamente tenga efecto en páginas del dominio de Gmail. Las extensiones pueden optar por disponer de esta herramienta opcional pero en este caso su utilidad principal es dar un punto de almacenaje y administración de credenciales.

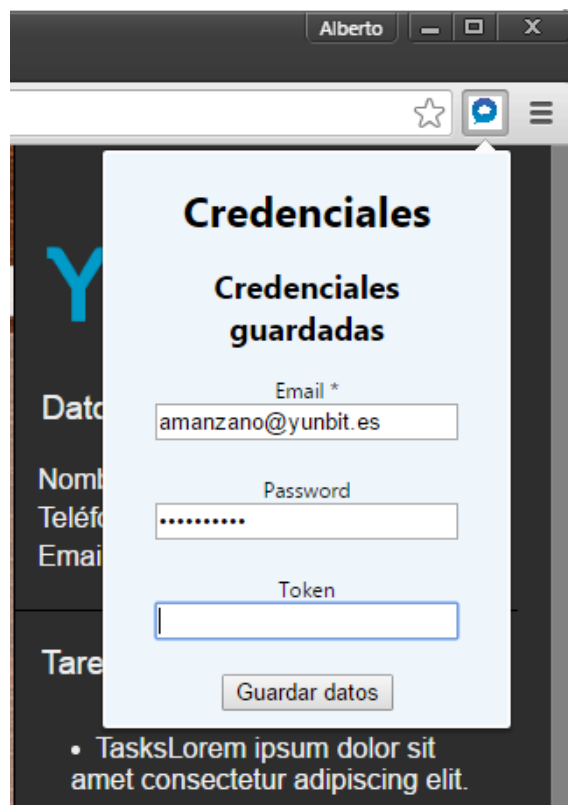


Figura 29. Action page extensión

- **Background:** Lo que es denominado el background *page* o página en segundo plano, es una página web que mantiene la lógica de la extensión suele ser considerado el núcleo de la lógica de la extensión. Existen dos tipos de background page en función del tipo de extensión que se quiera desarrollar. Persistente, como el nombre indica, son procesos de Google Chrome que siempre están abiertos, ejecutándose de manera continua. Páginas de eventos, son extensiones que se inician y se cierran cuando son necesitadas, suelen ser extensiones ligeras que no requieren de interacciones de usuarios. En este caso se ha optado por un background persistente que siempre está abierto para minimizar el tiempo que tarda la extensión en inyectarse en la interfaz de Gmail.

Desde el background se dispone de la biblioteca de funciones propias de las extensiones.

Se dispone de una funcionalidad para el paso de mensaje entre las distintas capas de las extensiones, lo veremos a continuación debido a su importancia.

Por otro lado existen funciones para el almacenaje de datos persistentes que han sido utilizadas para el guardado de las credenciales del usuario.

Podemos distinguir una serie de funcionalidades de las que se encarga este módulo:

- Detección de una *URL* del dominio de Gmail para la activación de la action page antes mencionada.
 - Detección de cambios en las pestañas de Gmail para la inyección del panel correcto.
 - Realización de peticiones AJAX para la obtención o inserción de datos en el servidor de Yunbit.
- **Content Script:** Capa intermedia entre el background y los elementos DOM de la página web que permite la interacción con los elementos de la página web que se está visualizando, es un Javascript ejecutado en el contexto de la página web y cargado desde el navegador.

Cuando hablamos de los content-scripts hablamos de mundos aparte (*isolated worlds*) que poseen una copia de los elementos DOM de la página web pero haciendo que su referencia a *window* no tenga porqué ser la misma que los Javascript que ejecuta directamente la página web.

Desde esta capa nos podemos comunicar con el background a través a través de un sistema de paso de mensajes de tal manera que ya tendríamos de la capacidad de modificar todos los elementos DOM de la interfaz de Gmail, pero existe un gran problema con este último aspecto.

Gmail administra los elementos de su interfaz de manera, digamos automática, a base de identificadores y clases con nombres no representativos o fáciles de utilizar, lo que hace que trabajar con una interfaz que puede variar en cualquier momento y dependamos de hacer ingeniería inversa para localizar los elementos en la interfaz, no sea la mejor solución.

A partir de este punto es cuando se ha optado por hacer uso de una librería de código abierto llamada *Gmail.js*[18] que facilita en gran medida la interacción con los elementos de la interfaz de Gmail.

Esta librería de código abierto da una serie de herramientas para trabajar fácilmente con la barra de botones (*toolbar*) barra en la que se va a inyectar un botón y permite crear ventanas con el diseño de Google. Se ha llevado a cabo una modificación de la librería para adecuarse al diseño de esta extensión y permitir la inyección de *sidebars* o barras laterales con información.

Para hacer uso de esta librería como el autor indica es necesario hacer uso de JQuery y de la nueva capa que vamos a ver en detalle a continuación esto es debido a que esta librería únicamente puede funcionar si se encuentra inyectada en la cabecera del HTML que va a hacer su uso, tal y como indica su autor.

- **Injected Script:** Capa de menor nivel por ser inyectada directamente en el *header* del HTML de la interfaz de Gmail, desde esta capa podemos utilizar la librería de

Gmail.js e interactuar con la interfaz de Gmail para inyectar el diseño de la extensión.

Al no encontrarnos dentro del ecosistema de la extensión propiamente dicho, la comunicación entre las capas superiores y esta se ha de realizar mediante la gestión de eventos Javascript.

Al no tener acceso a las librerías de almacenaje de datos que hace uso el *background* y por motivos de seguridad, las peticiones al servidor externo de Yunbit se realizan desde el background con un sistema de paso de mensajes entre las 3 capas que veremos a continuación, Figura 30.

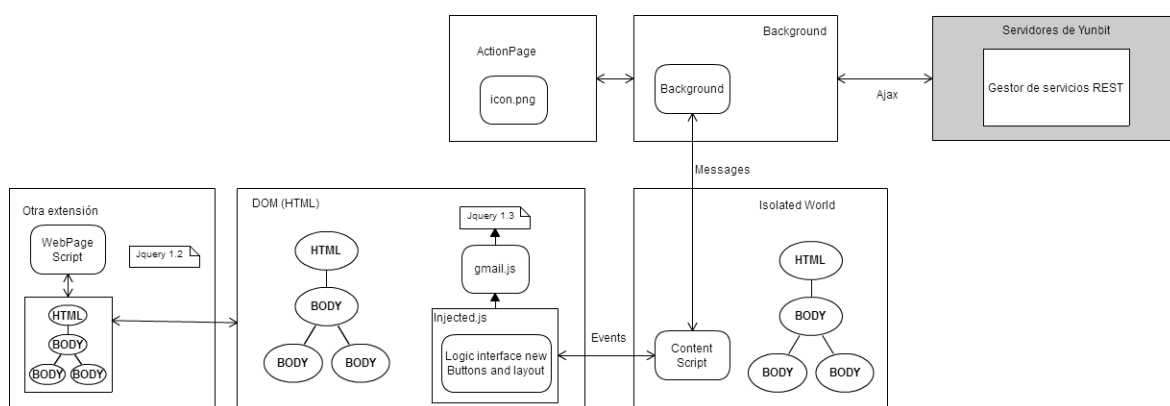


Figura 30. Arquitectura de la extensión

Para aclarar el funcionamiento del paso de mensajes se adjunta otro diagrama indicando como se produce el proceso para la inyección del primer panel, Figura 31.

Con primer panel, se hace referencia al diseño de la interfaz cuando se inyecta por primera vez, cuando el usuario se encuentra en la bandeja de correo por ejemplo.

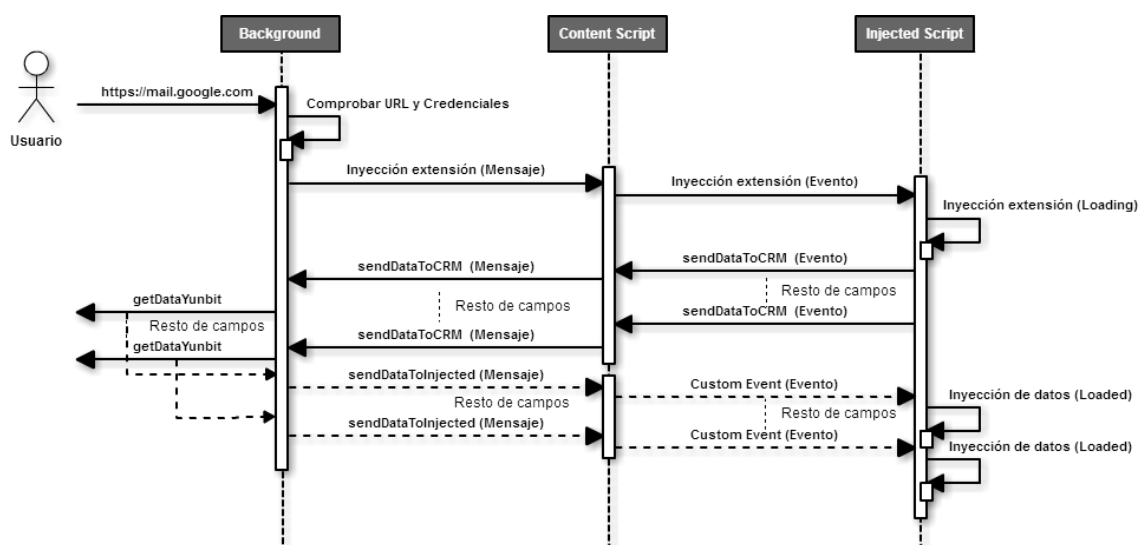


Figura 31. Diagrama de secuencia, inyección extensión

La extensión dispone de dos áreas informativas con datos de la plataforma de Yunbit, para entender mejor su funcionamiento así como su estructura se acompañan una serie de capturas de la extensión con datos de prueba.

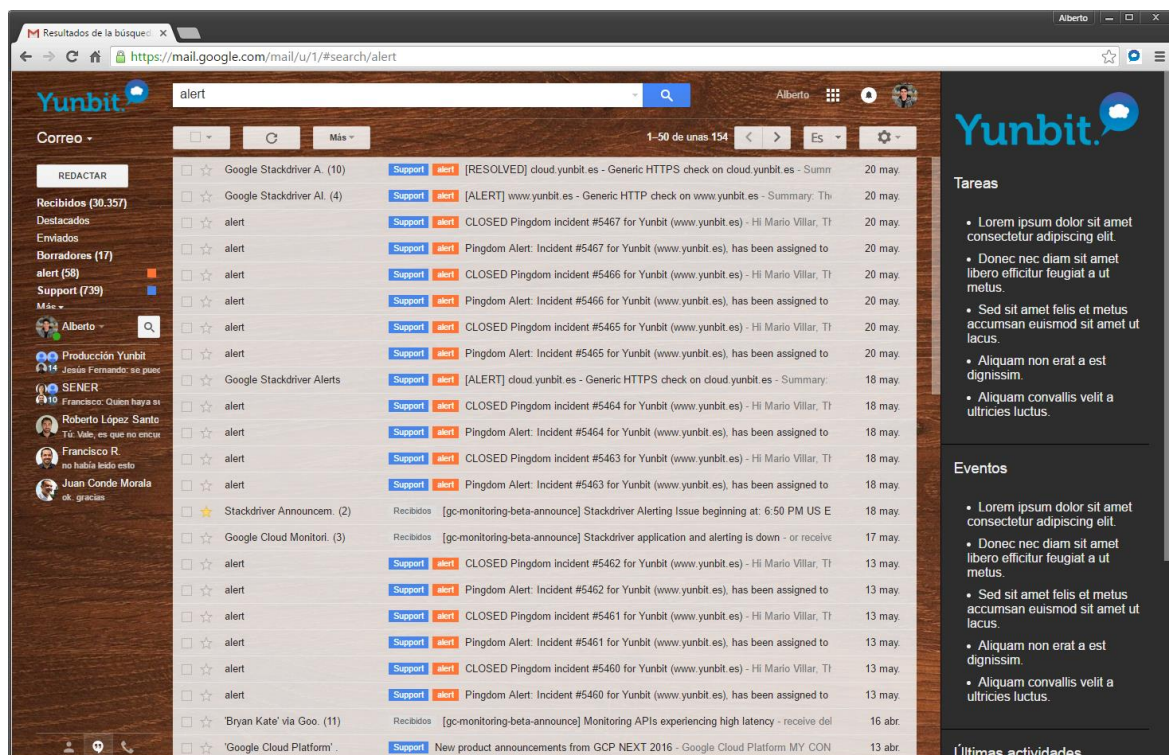


Figura 32. Bandeja de correo con panel de usuario

En esta primera figura del diseño de la extensión podemos apreciar un área lateral que ocupa en parte el ancho de la interfaz de Gmail, este área es lo que es denominado “marco Yunbit”, este marco es inyectado en la interfaz de Gmail una vez que esta da el aviso de que existen credenciales y se ha realizado un evento de navegación o de actualización.

Este primer panel muestra información relativa al usuario que hemos insertado en las credenciales, información que proviene de la plataforma de Yunbit como son las tareas, eventos y últimas actividades realizadas. Estos aspectos permiten a los usuarios de Yunbit tener de una manera centralizada la información de su cuenta sin necesidad de acceder al portal, si no que desde el propio correo se pueden gestionar diversas funcionalidades.

En esta área que es común al área que se genera al entrar en la visualización de un email en detalle, está compuesta por una serie de marcos.

Cada marco representa uno de los servicios REST del servidor de Yunbit del que se recogen los datos y de esta manera se pueden definir especializaciones de estos marcos en función de la necesidad de representación de los datos obtenidos. En este caso todos los marcos son del mismo tipo pero como veremos en el otro panel, se ha definido un marco de tipo tabla en lugar del tipo lista como el aquí mostrado.

Estos marcos tienen en común que cuando no se dispone del contenido que han de mostrar, muestran un *spinner* de Google, Figura 33, para indicar al usuario que se está

cargando el contenido y cuando finalmente es obtenido, se reemplaza por el marco con el formato solicitado.

Como las peticiones AJAX son peticiones asíncronas es de vital importancia para los usuarios el saber que no se muestran los datos porque estos se están obteniendo.

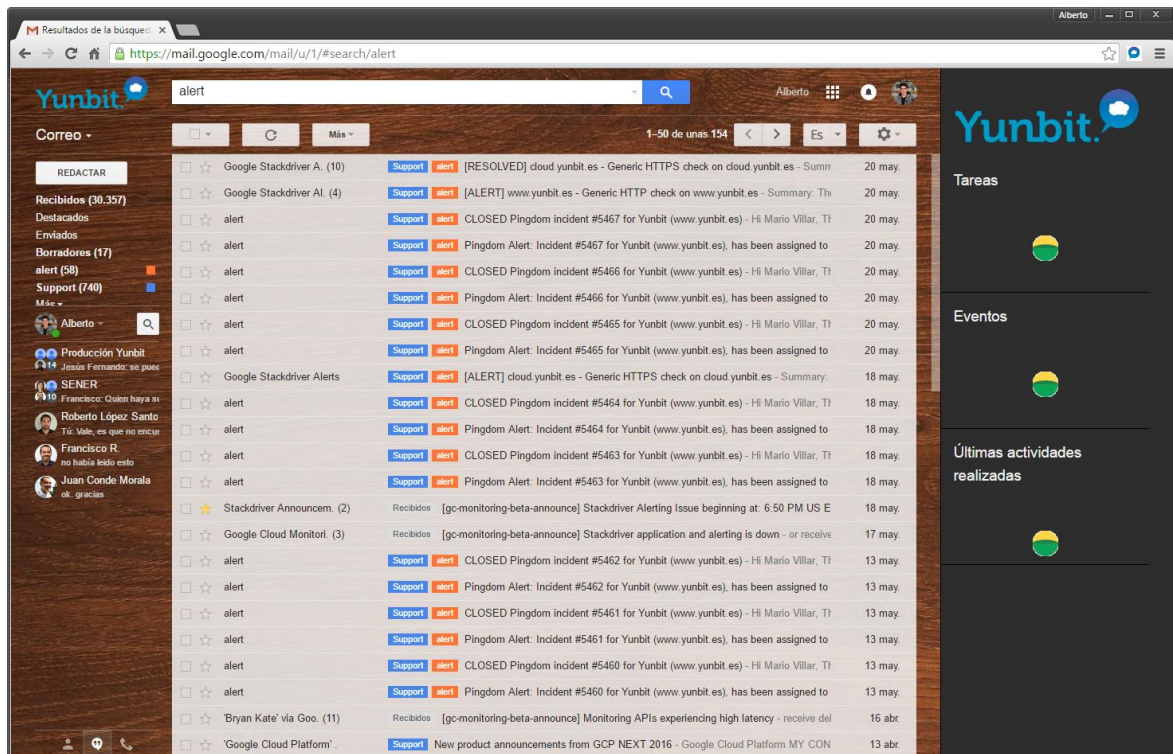


Figura 33. Bandeja de correo con panel de usuario (Loading)

Para entender mejor el funcionamiento de estos marcos, se plantea el siguiente escenario:

1. Injected Script recibe el orden de inyectar la extensión a través de un evento desde el Content Script.
2. Se crean los marcos con los spinners y se realiza la solicitud del dato a Content a través de un evento.
3. Content tras haber contactado con Background y haber obtenido el contenido, lo reenvía a Injected haciendo uso de un evento con el nombre del campo a sobrescribir.
4. Injected recibe un evento de sustitución de spinner por contenido, en función del nombre de este evento se reemplaza un campo u otro.

Se ha decidido incluir un segundo panel, Figura 34, que se muestra cuando nos encontramos dentro de un email, este panel de carácter informativo muestra datos relativos al sujeto que ha enviado dicho email. Estos datos son extraídos a través de servicios desde Yunbit.

A su vez cuando nos encontramos dentro de un email, aparece en el toolbar un nuevo botón que implementa la función de sincronización del email, es decir, el envío del contenido del email a la plataforma de Yunbit para su sincronización.

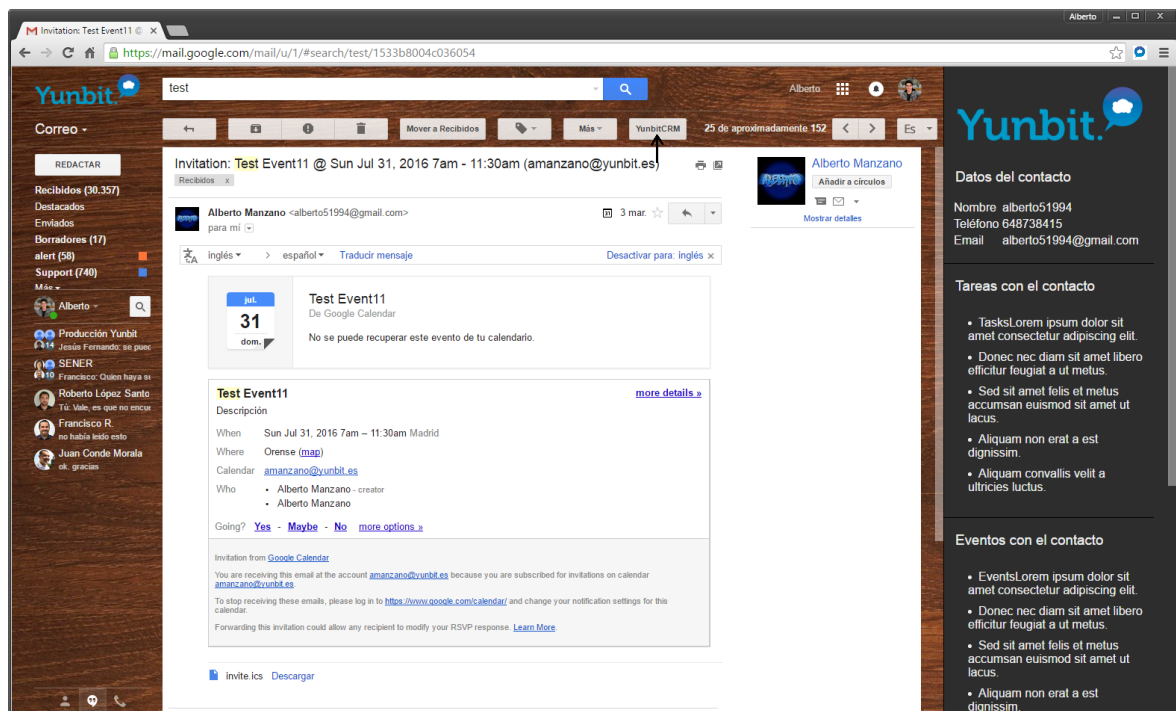


Figura 34. Visualización de correo con panel

Se obtienen datos tales como datos personales, tareas, eventos y oportunidades del contacto, a su vez la gestión de este panel se realiza a través del sistema de marcos, la estructura del marco de Datos del contacto varía frente al resto. Estos marcos, son fácilmente adaptable a las necesidades de visualización y permiten en un futuro implementar diferentes estilos y diseños.

En relación con el nuevo botón, que únicamente es inyectado en la interfaz cuando el usuario se encuentra dentro del email, hace que se muestre un formulario en una ventana emergente dentro de la interfaz de Gmail, con los datos a rellenar por el usuario para la vinculación dentro de la plataforma de Yunbit.

Estos campos del formulario, Figura 35, a pesar de que no funcionan como marcos, siguen teniendo la capacidad de mostrar un spinner mientras se cargan los valores de la plataforma de Yunbit.

Actualmente debido a que no están implementados los servicios en la parte de Yunbit, se simula el envío notificando al usuario que el email seleccionado con los valores del formulario ha sido sincronizado.

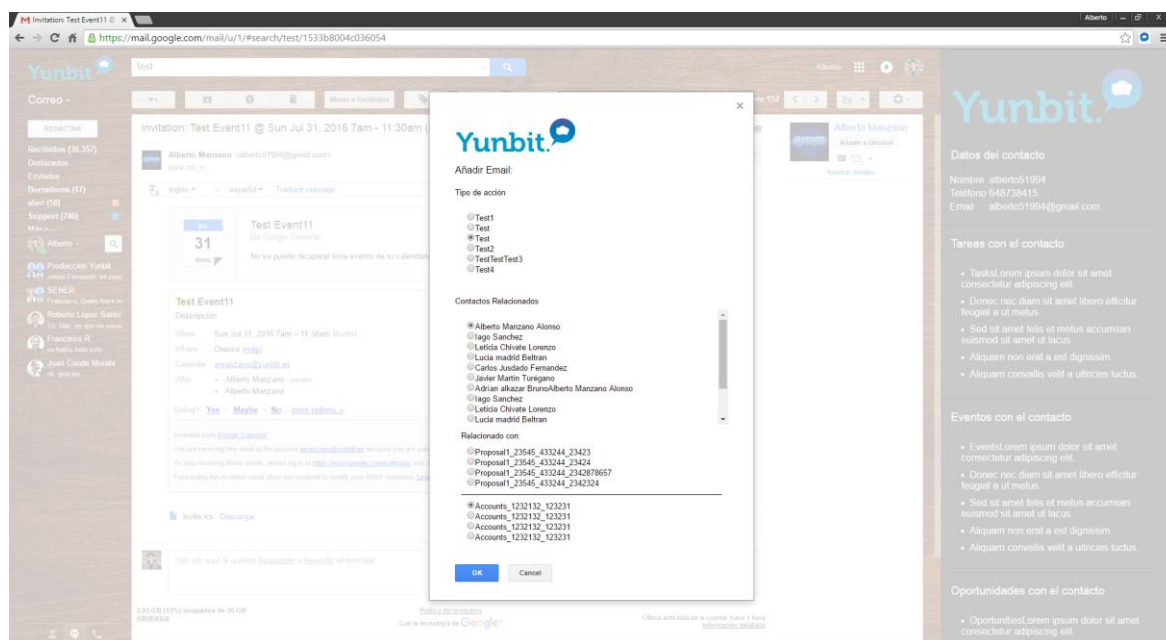


Figura 35. Formulario vinculación email

Un aspecto muy importante a tener en cuenta durante el desarrollo de extensiones a nivel de seguridad, es el CSP (*Content Security Policy*) y CORS (*Cross-Origin Resource Sharing*).

- **CSP:** Dentro del manifest se puede declarar una barrera adicional de protección para las peticiones que realice la extensión, evitando así evaluar contenidos traídos a través de solicitudes a servidores externos, evaluar códigos Javascript *in-line*, etc. Con esta herramienta se permite un mayor control de los ficheros que se ejecutan y dominios a los que se realizan llamadas, se puede considerar un mecanismo de listas de recursos permitidos y prohibidos. En este caso se han habilitado las peticiones a los servidores de Yunbit.
- **CORS:** En ocasiones y más exactamente esta ocasión, las extensiones necesitan realizar peticiones a servidores remotos. Estas peticiones están limitadas por la política de *same-origin* que de manera estándar solo permite acceder a contenidos que estén almacenados bajo el mismo dominio, para permitir la realización de estas llamadas durante la fase de desarrollo es necesario permitir este tipo de peticiones desde el servidor que las atiende añadiendo un *header* de permiso. Cuando la extensión ya se encuentra empaquetada y sin estar en desarrollo local, tan solo es necesario dar permisos a los dominios que se soliciten en el manifest siempre y cuando concuerden con el CSP en caso de estar declarado.

7 Conclusiones y trabajo futuro

Tras la realización de este proyecto, Yunbit dispone de la estructura de un sistema de sincronización con Google fácilmente personalizable a las necesidades de los clientes y fácilmente extensible con nuevas funcionalidades que puedan requerir en el futuro.

A medida que he ido trabajando con la librería de Google para PHP me he dado cuenta de la necesidad que tiene la comunidad de liberar proyectos de este tipo para facilitar la creación de nuevos proyectos por la falta de documentación o de ejemplos complejos.

He cometido los errores comunes en este tipo de proyectos con Google por lo que he podido apreciar en Internet, hasta que finalmente he desarrollado la capacidad de moverme por las herramientas y documentación que Google proporciona con soltura, obteniendo la capacidad de abstracción necesaria para entender el funcionamiento de todo lo anterior visto como un conjunto.

Las herramientas desarrolladas son útiles para los usuarios de la plataforma de Yunbit y fácilmente extensibles con nuevas funcionalidades como podrían ser:

- Creación de los servicios REST en la plataforma Yunbit para dar soporte a las funcionalidades de la extensión de Google Chrome para la interfaz de Gmail.
- Creación de un cargador de ficheros automático desde dentro de la plataforma de Yunbit para Google Drive, evitando así el tener que subir el documento a Google Drive de manera manual para posteriormente obtener los metadatos en la plataforma de Yunbit.
- Mejorar el soporte al sistema de eventos de Google rediseñando la estructura actual de eventos de la plataforma Yunbit para adecuarse a los RFC de eventos y fechas.
- Añadir más funcionalidad a la extensión de Google Chrome para fomentar el uso y la instalación por parte de los usuarios.
- Optimizar y mejorar los tiempos de ejecución de algunas tareas de sincronización.
- Hacer la extensión de Google Chrome redimensionable o adaptable a pantallas en las que su visualización no sea la más adecuada.

Por último quedaría realizar más pruebas de los módulos desarrollados antes de hacerlos públicos a todas las plataformas y realizar observaciones del uso que los usuarios dan a la herramienta para mejorar aquellos puntos más utilizados.

Referencias

- [1] Yunbit S.L., 15 Enero 2016
<http://www.yunbit.es>
- [2] Google OAuth, 22 Enero 2016
<https://developers.google.com/identity/protocols/OAuth2>
- [3] JQuery
<https://jquery.com>
- [4] Gmail.js, 10 Marzo 2016
<https://github.com/KartikTalwar/gmail.js/tree/master>
- [5] Google Client PHP API, 25 Enero
<https://developers.google.com/api-client-library/php>
- [6] Wunderlist
<https://www.wunderlist.com/es>
- [7] Syncro SVN Client
<http://www.syncrosvnclient.com>
- [8] PhpStorm
<https://www.jetbrains.com/phpstorm>
- [9] WampServer
<http://www.wampserver.com/en>
- [10] Navicat
<https://www.navicat.com>
- [11] Filezilla
<https://filezilla-project.org>
- [12] Putty
<http://www.putty.org>
- [13] Google Calendar
<https://calendar.google.com/calendar>
- [14] Google API Console, 19 Enero 2016
<https://console.developers.google.com>
- [15] Google Maps, 17 Mayo 2016
<https://www.google.es/maps>

- [16] Google Contacts, 18 Mayo 2016
<https://www.google.com/contacts>
- [17] Google Data API, 25 Febrero 2016
<https://developers.google.com/gdata/>
- [18] GData Protocol, 5 Abril 2016
<https://developers.google.com/gdata/docs/2.0/reference>
- [19] Gmail.js, 20 Febrero 2016
<https://github.com/KartikTalwar/gmail.js/tree/master>
- [20] Ebsta para Salesforce
<https://chrome.google.com/webstore/detail/ebsta-for-salesforce/gemcgnkghpnfbmlfimdbdgfepcgenphf?hl=en-GB>
- [21] Strike
<https://chrome.google.com/webstore/detail/strike-for-gmail-beta/pofgjjnmencehbdejkdapopepabpej?hl=en>
- [22] Right Inbox
<https://chrome.google.com/webstore/detail/right-inbox-for-gmail/mflnemhkomgploogccdmcloekbloobgb?hl=enrightinbox>
- [23] El camelizer
<https://chrome.google.com/webstore/detail/the-camelizer/ghnomdcacenbmilgjigehppbamfndblo?hl=enrightinbox>
- [24] Certificado (ANEXO C)
<http://curl.haxx.se/ca/cacert.pem>

Glosario

API	<i>Application Programming Interface</i>
AJAX	<i>Asynchronous Javascript and XML</i>
CSP	<i>Content Security Policy</i>
CRM	<i>Customer Relationship Management</i>
CORS	<i>Cross-Origin Resource Sharing</i>
DOM	<i>Document Model Tree</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
JSON	<i>Javascript Object Notation</i>
TPL	<i>Template</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>Extensible Markup Language</i>
XMLNS	Espacio de nombres XML

Anexos

A Creación de proyectos en la consola de desarrolladores de Google

Para la creación de un nuevo proyecto de Google que haga uso de *APIs* es necesario en primer lugar visitar la Google API Console[14] y disponer de una cuenta de Google.

Desde cualquier parte de la aplicación tendremos acceso al desplegable de selección de proyectos desde el cual podremos crear el nuevo proyecto a utilizar, a través de este seleccionable se abrirá un formulario donde rellenando únicamente el nombre del proyecto se creará automáticamente.

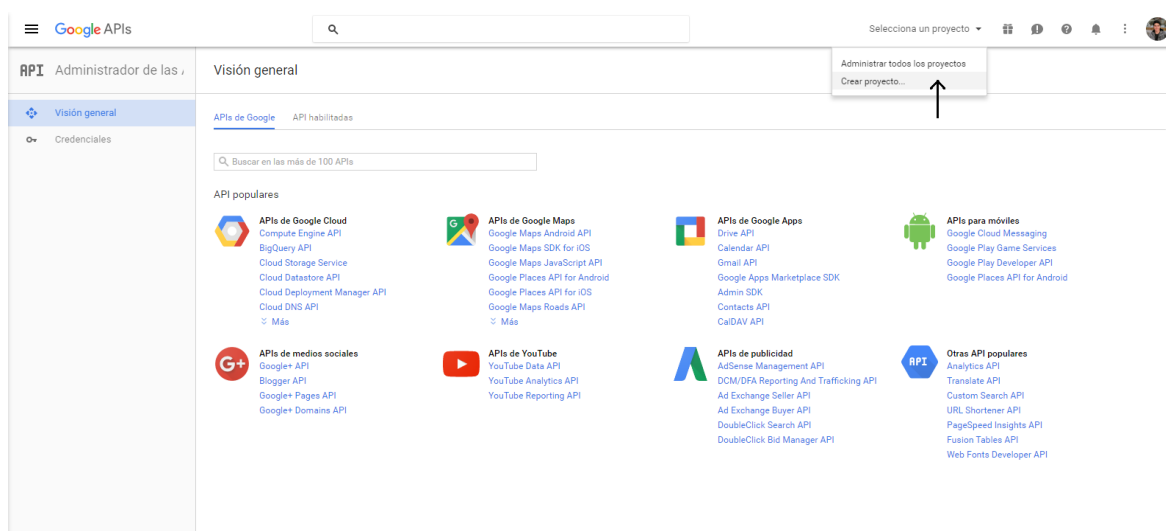


Figura 36. Google Console *Developers* (Vista general)

Una vez creado el proyecto procederemos a la autorización de uso de las *APIs* de las que vayamos a hacer uso de manera manual seleccionándolas del listado proporcionado en la Figura 36.

B Creación y administración de credenciales de proyectos para aplicaciones web

Para realizar este paso es necesario haber realizado los pasos mencionados en el ANEXO A para la creación de un proyecto.

En este caso vamos a trabajar con credenciales para aplicaciones web, por ser las utilizadas en este TFG, para ello accedemos a la vista de la Figura 37 de la Google API Console[14].

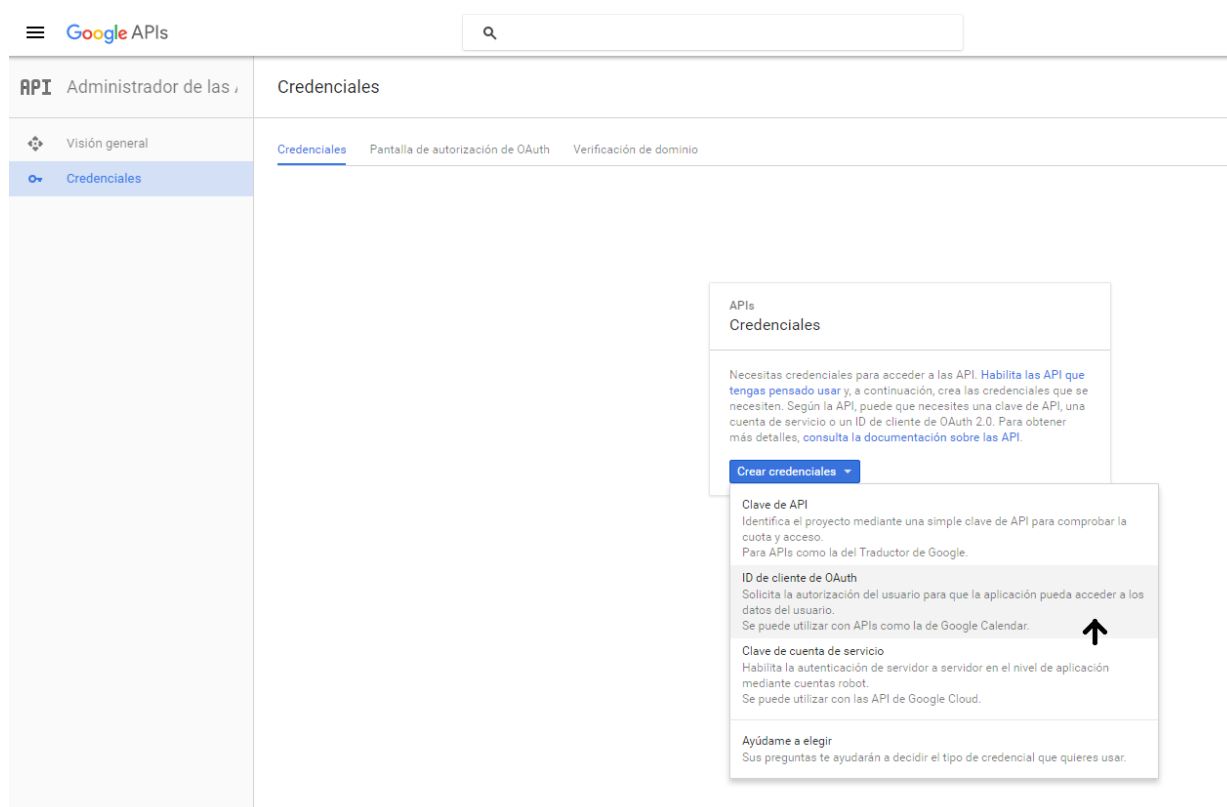


Figura 37. Google Console Developers (Credenciales)

A continuación en primer lugar se solicitará al administrador que cumplimente un formulario de configuración de la pantalla de autorización OAuth, Figura 38, esta configuración se aplica a la pantalla que se mostrará a los usuarios de la aplicación cuando den autorización de uso de los servicios requeridos.

Los datos obligatorios a cumplimentar son:

- **Dirección de correo electrónico:** En caso de producirse un error durante el proceso de autorización del usuario, Google muestra de manera automática la dirección de correo electrónico de la plataforma para que los usuarios puedan ponerse en contacto con la misma dando datos del error producido.

- **Nombre del producto mostrado a los usuarios:** Este nombre es una referencia clara junto con el logotipo de la empresa para el usuario para saber a quién está dando permisos de manera exacta.

Existe una serie de campos adicionales que se pueden rellenar si se desea, no obstante es muy recomendable su cumplimentación con el fin de dar más seguridad al usuario final durante el proceso de autenticación de permisos.

Credenciales


Credenciales
Pantalla de autorización de OAuth
Verificación de dominio

Dirección de correo electrónico ?

Nombre de producto mostrado a los usuarios

URL de página principal (Opcional)


URL de logotipo de producto (Opcional) ?



Así es como verán los usuarios finales tu logotipo
Tamaño máximo: 120 x 120 píxeles

URL de la Política de Privacidad (Opcional)

URL de las Condiciones de Servicio (Opcional)



La pantalla de autorización se muestra a los usuarios cuando solicitas acceso a sus datos privados mediante tu ID de cliente. Se muestra para todas las aplicaciones registradas en este proyecto.

Debes proporcionar una dirección de correo electrónico y un nombre de producto para que OAuth funcione.

Figura 38. Configuración de pantalla OAuth

Por último pero muy importante, tenemos que definir las URLs de redirección que vamos a utilizar en nuestras aplicaciones, esto es, una entrada por cada proyecto que haga uso de este servicio y tenga una *URL* diferente.

Estas URLs hay que añadirlas en el campo “*URLs de redireccionamiento autorizados*” de la Figura 39, como únicamente trabajamos con solicitudes desde un servidor web, este será el único campo que necesitaremos rellenar para poder trabajar con la plataforma, a su vez cabe destacar que han de ser URLs sin rutas relativas.

Credenciales

←
Descargar JSON
Restablecer el secreto
Eliminar

ID de cliente para Web

ID de cliente	241423086942-plqdce2okeq0ljemjb60ronqndsquv6t.apps.googleusercontent.com
Secreto de cliente	KtjiSG-1VVRjQTmhnYbIE3NI
Fecha de creación	17 may. 2016 11:23:31

Nombre

Muestra TFG

Restricciones

Introduce los orígenes de JavaScript, los URI de redireccionamiento o ambos

Orígenes de JavaScript autorizados

Para su uso en las solicitudes de navegador. Se trata del URI de origen de la aplicación cliente. No puede contener caracteres comodín (`http://*.example.com`) ni una ruta (`http://example.com/subdir`). Si utilizas un puerto no estándar, deberás incluirlo en el URI de origen.

`http://www.example.com`

URLs de redireccionamiento autorizados

Para usarse con las solicitudes de un servidor web. Es la ruta de la aplicación a la que se redirecciona a los usuarios después de autenticarse en Google. A dicha ruta se añadirá el código de autorización de acceso. Debe tener un protocolo. No puede incluir fragmentos de URL ni rutas relativas. No puede ser una dirección IP pública.

`http://ppe-cloud-adysagroup.geanetondemand.com/IntranetHome/showURL?url=/ADYSAGROUP_INTRANET/GEA-WEBCONTROL/PersonalInfo.php&title=Datos+personales` ✕

`http://www.example.com/oauth2callback`

Guardar Cancelar

Figura 39. Configuración de credenciales

Una vez que se haya definido la configuración de las credenciales, se ha de descargar el fichero JSON con la información nueva y transferirlo a los servidores o lugares donde se hacer uso de la plataforma en el mismo directorio que la instalación de la librería de Google, este fichero JSON es el denominado secreto de cliente.

Esto es necesario ya que si no concuerdan las direcciones de redireccionamiento definidas en la Google API Console con las URLs en el JSON utilizado, Google mostrará un error de desajuste entre las URLs en la pantalla de autorización OAuth.

C Instalación del proyecto en local

Para llevar a cabo el proceso de instalación del proyecto de manera local, se han de seguir los siguientes pasos:

- Para la instalación de la librería PHP de Google se recomienda encarecidamente hacer uso de un administrador de dependencias PHP como *Composer* que realizará de manera automática el proceso de instalación y configuración de la librería con sus correspondientes dependencias.
Otro aspecto importante de la librería es descargar la versión 1.1.6 que hace uso de la versión de PHP 5.4 presente en la plataforma de Yunbit.
- Además hace falta configurar la extensión de "php_openssl" en la configuración de WampServer para que se admitan las peticiones que realizamos a Google, a su vez es necesario modificar el documento "php.ini" de la versión de PHP que estemos utilizando descomentando la línea correspondiente al dll de "openssl".
- Para el correcto funcionamiento de la aplicación hay que tener en cuenta que las peticiones curl que se realizan al servidor de Google, necesitan comprobar que el certificado de Google es correcto, para ello es necesario añadir un *bundle*, conjunto, de direcciones certificadoras que validen dicha dirección. Este fichero no viene de forma automática con open_ssl por lo que hay que descargarlo [24].
De manera adicional para que WampServer reconozca este bundle, tenemos que modificar los ficheros de inicio "php.ini" tanto de Apache como de la instalación de Wamp, añadiendo la siguiente variable de configuración:

curl.cainfo = "path\cacert.pem"

Esto permitirá a la extensión de open_ssl que cuando se realice una petición curl pueda verificar el certificado CA.